



Universidad  
Carlos III de Madrid  
www.uc3m.es

## ***TESIS DOCTORAL***

# ***Modelo de Requisitos Orientado al Reúso Efectivo (MORORE)***

**Autor:**

**Omar Armando Manuel Hurtado Jara**

**Directores:**

**Dr. Gonzalo Génova Fuster**

**Dra. Anabel Fraga Vázquez**

**Tutor:**

**Dr. Gonzalo Génova Fuster**

**DEPARTAMENTO DE INFORMÁTICA**

**Leganés, junio 2017**



## TESIS DOCTORAL

### MODELO DE REQUISITOS ORIENTADO AL REÚSO EFECTIVO (MORORE)

**Autor:** *Omar Armando Manuel Hurtado Jara*

**Directores:** Dr. Gonzalo Génova Fuster

Dra. Anabel Fraga Vázquez

Firma del Tribunal Calificador:

Firma

Presidente: (Nombre y apellidos)

Vocal: (Nombre y apellidos)

Secretario: (Nombre y apellidos)

Calificación:

Leganés, de junio de 2017

*“Lo que con mucho trabajo se adquiere, más se ama.”*

Aristóteles

## AGRADECIMIENTOS

Quisiera mencionar en primer lugar a Dios, con él todo es posible.

Después del largo camino en el desarrollo de esta tesis, son incontables las personas a las que debo agradecer su apoyo. Aquellas que estuvieron ahí, en los altos y bajos, que de diferentes formas colaboraron para que este trabajo fuera posible. A todos les estoy muy agradecido.

Quiero agradecer a mi familia: mis hijos Nadia, Bruno y Diego que con tan solo existir me dieron la fuerza para seguir. A mi querida esposa Rossy que me acompaña en el andar de la vida. A mis padres Emma y Manuel, que me dieron la vida y su esfuerzo, junto con mis hermanos, para sacarme adelante. No puedo dejar de mencionar a mi Tía Clara, quien se convirtió en una segunda madre en mi estadía en España, también mencionar a mis primos Vicente, Sara y Daniel. A mis primos de Piura, Daniel y Christian y sus respectivas familias por siempre estar acompañándome.

Quiero agradecer a la Universidad de Piura, que me dio la oportunidad de poder realizar los estudios. Y a todos sus miembros de la Facultad de Ingeniería, qué como una familia, siempre estuvieron pendientes de este trabajo. A los miembros de mi área por la gran amistad que tenemos: Alejandro, Eugenio, Mario, Erick y Jorge.

También agradezco profundamente a Gonzalo y Juan, de quienes guardo una gran admiración y de los cuales he aprendido mucho para mi desarrollo, y vida profesional y personal. Asimismo, a todos los miembros del grupo Knowledge Reuse Group: mi amiga y codirectora Anabel, a Eduardo, a Diego, a Mónica, a Valentín, a Yorgos, a Karina, a Jorge, a Sonia, a Maricruz, a Vicente, a Isidro, a Julian...; y de The Reuse Company encabezados por José Miguel (JM) y Emiliano; que hicieron que mi estadía en España fuera muy agradable.

También quiero agradecer de manera especial a Manuel López (Manolo) entrañable amigo de la familia. A Vladimir, por la gran amistad que tenemos. A mi compadre Guillermo por tantos buenos momentos compartidos. A todos los incontables amigos de “300”: Yimy, Johann, Joaquín, Hugo, Pedro y un largo etc., quienes hicieron sentirme como en casa.

Sé que obvio a muchas personas que de forma directa o indirecta colaboraron para que pueda culminar este trabajo, sé que me conocen y me sabrán dispensar.

A todos en general, sinceramente les agradezco de todo corazón.



## ABSTRACT

The main problems faced by the software engineering have their origins in requirements engineering. Many studies have shown the importance of the requirements engineering in the software development process, especially the phases of specification and requirements management. Consequently, today many researchers focus their efforts on improving this discipline and therefore the software development process.

There are many proposed solutions to improve the requirements engineering: standards, tools, methods and techniques. Among these, one of the most innovative proposals is the application of reuse techniques in the requirements engineering.

In concordance with this trend, this thesis aims to develop a model for the reuse of requirements called: Model of Requirements Oriented to Effective Reuse (MORORE, for its acronym in Spanish).

MORORE allows the representation of three levels of software assets (also called software artifacts) for the reuse of requirements: individual representation of requirements, representation of sets of requirements (requirement patterns) and representation of requirements types structures (requirements types structures patterns). Also in this thesis, we consider the definition and implementation of metrics for quality control, of products or reusable assets, within indexing and retrieval processes using MORORE.

The level of individual representation of requirements allows the correct and complete specification of a requirement for its reuse in a new project. This specification includes a collection of descriptive attributes, the association with other requirements and external elements. Also, this level allows the parameterization of requirements. This parameterization aims to ease the individualization of the requirement's variable data, in order to facilitate its adaptation to a new project.

For the level of the requirements pattern, it is presented a representation of a solution as a set of interrelated requirements which is a proven solution in a recurring problem in the field of requirements engineering. This level includes the representation of the descriptive attributes of the pattern, the specification of the constituent elements ("requirement", assuming, if necessary, all the characteristics of individual requirement) and the associations between the pattern and other external elements.

For the level of requirements types structures patterns, we consider the representation of a set of types of requirements organized in a general structure. This type of pattern includes a set of descriptive attributes of the structure of types to allow the location and learning to be used. This level also includes the representation of other associated elements to the structures: files, web pages, etc.

In addition, it is proposed the application of a quality control for an effective reuse of reusable products at each one of the model's levels. This quality control consists in the definition of a set of factors and metrics to guarantee a minimum quality of reusable assets at each level of MORORE.

Within the scope of this thesis, it is also considered the definition of a reusable-assets indexing and retrieval process for each one level of the model. This process describes the steps the software assets must follow at each level of MORORE for an efficient reuse. These steps describe the software asset cycle from its conception or definition, to its adaptation to a new project at the time of reuse.

We believe that the application of MORORE will contribute to improvements in the requirements engineering effectiveness and efficiency and therefore in the software development process. Specifically, with the implementation of MORORE there will be improvements in the quality specification by engineers and it will be achieved a decrease in the development process time and in the resources used in the requirements engineering.

## RESUMEN

Los principales problemas que afronta la ingeniería de software tienen su origen en la ingeniería de requisitos. Muchos estudios han demostrado la importancia que tiene la ingeniería de requisitos para el proceso de desarrollo de software, especialmente las fases de especificación y gestión de los requisitos. En consecuencia, en la actualidad muchos investigadores orientan sus esfuerzos al mejoramiento de esta disciplina y por consiguiente del proceso de desarrollo de software.

En la actualidad, existen muchas propuestas de solución para mejorar la ingeniería de requisitos: estándares, herramientas, métodos y técnicas. Entre éstas, una de las más novedosas propuestas es la aplicación de técnicas de reúso al ámbito de la ingeniería de requisitos.

En concordancia a esta tendencia de aplicación del reúso al ámbito de la ingeniería de requisitos, esta tesis doctoral propone desarrollar un modelo para el reúso de requisitos denominado: *Modelo de Requisitos Orientado al Reúso Efectivo (MORORE)*.

MORORE permite la representación de tres niveles de activos software (también llamados artefactos software) para el reúso de requisitos: representación individual de requisitos, representación de conjuntos de requisitos (patrones de requisitos) y representación de estructuras de tipos de requisitos (patrones de estructuras de tipos). Asimismo, en la presente tesis doctoral, consideramos la definición y aplicación de métricas para el control de calidad de los productos o activos reutilizables, dentro de un proceso de indexación y recuperación, utilizando MORORE.

El nivel de representación individual de requisitos permite la especificación correcta y completa de los requisitos para su reúso en un proyecto nuevo. Esta especificación incluye una colección de atributos descriptivos, la asociación con otros requisitos y con elementos externos. Asimismo, este nivel permite la parametrización de requisitos. Esta parametrización consiste en facilitar la particularización de los datos variables de un requisito para facilitar su adaptación a un proyecto nuevo al momento de reusarlo.

Para el nivel del patrón de requisitos, planteamos la representación de un conjunto-solución de requisitos interrelacionados que se ha demostrado resuelven un problema recurrente del ámbito de la ingeniería de requisitos. Este nivel incluye la representación de los atributos descriptivos del patrón, la especificación de los elementos constitutivos (“requisito”, asumiendo, si es el caso, todas las características del requisito individual) y las asociaciones entre el patrón y otros elementos externos.

Para el nivel de patrón de estructuras de tipos de requisitos, consideramos la representación de un conjunto de clasificaciones de requisitos organizados en una estructura general. Este tipo de patrón incluye un conjunto de atributos descriptivos de la estructura de tipos que permitan la ubicación y aprendizaje para su uso. Este nivel también incluye la representación de otros elementos asociados a las estructuras: archivos, páginas web, etc.

Además, proponemos la aplicación de un control de calidad para el reúso efectivo de los productos reusables de cada uno de los niveles del modelo. Este control consiste

en la definición de un conjunto de factores y métricas que garanticen un mínimo de calidad de los activos reusables de cada nivel de MORORE.

Dentro del ámbito de la presente tesis doctoral también consideramos la definición de un proceso de indexación y recuperación para los activos reusables de cada nivel del modelo. Este proceso describe los pasos que deben de seguir los activos software de cada nivel de MORORE para su reúso eficiente. Estos pasos describen el ciclo del activo software desde su concepción o definición hasta su adecuación a un proyecto nuevo al momento de reusarlo.

Consideramos que la aplicación de MORORE contribuirá a la mejora en efectividad y eficiencia de la ingeniería de requisitos y por tanto del proceso de desarrollo de software. En forma específica con MORORE habrá mejoras en la calidad de la especificación por parte de los ingenieros. Asimismo, con la aplicación de MORORE también obtendremos la disminución en el tiempo de los procesos de desarrollo respectivo y por consiguiente de recursos utilizados en el proceso de la ingeniería de requisitos.

# ÍNDICE

## MODELO DE REQUISITOS ORIENTADO AL REÚSO EFECTIVO (MORORE) ..... 1

### 1. INTRODUCCIÓN ..... 19

1.1 CONTEXTO DE LA INVESTIGACIÓN ..... 19

1.2 DESCRIPCIÓN DEL PROBLEMA ..... 20

1.3 IMPORTANCIA DEL PROBLEMA ..... 23

1.4 HIPÓTESIS DE INVESTIGACIÓN..... 25

1.5 OBJETIVOS ..... 25

1.6 APROXIMACIÓN A LA SOLUCIÓN ..... 26

1.7 VALIDEZ DE LA SOLUCIÓN ..... 29

1.8 METODOLOGÍA ..... 29

1.8.1 Revisión bibliográfica..... 30

1.8.2 Planteamiento de la hipótesis ..... 31

1.8.3 Selección de las fuentes ..... 31

1.8.4 Obtención de los modelos existentes en las fuentes ..... 31

1.8.5 Definición del modelo..... 32

1.8.6 Evaluación de la solución..... 32

1.9 LIMITACIONES Y AMENAZAS DE LA INVESTIGACIÓN ..... 33

1.10 ESTRUCTURA DE LA TESIS DOCTORAL ..... 33

### 2 ESTADO DE LA CUESTIÓN ..... 36

2.1 INGENIERÍA DE REQUISITOS..... 36

2.1.1 Definiciones de ingeniería de requisitos..... 37

2.1.2 Proceso de la ingeniería de requisitos..... 40

2.2 REÚSO..... 41

2.2.1 Definición de reúso de software ..... 41

2.2.2 Beneficios del reúso de software ..... 42

2.2.3 Dificultades del reúso de software..... 43

2.2.4 Activos Software reusables ..... 43

2.3 INDEXACIÓN Y RECUPERACIÓN DE LA INFORMACIÓN ..... 44

2.3.1 Modelos de representación de la información..... 45

2.3.2 Modelos de recuperación de la información ..... 46

2.4 PATRONES..... 47

2.4.1 Definiciones de patrón..... 47

2.4.2 Características de un patrón ..... 49

2.4.3 Descripción de un patrón ..... 49

2.4.4 Orientaciones de aplicación de un patrón ..... 50

2.5 CALIDAD EN LA INGENIERÍA DE SOFTWARE ..... 51

2.5.1 Importancia de la calidad de requisitos ..... 52

2.6	ESTÁNDARES DE REQUISITOS .....	53
2.6.1	<i>Métrica. Versión 3</i> .....	53
2.6.2	<i>Agencia Espacial Europea (ESA: European Space Agency)</i> .....	56
2.6.3	<i>ISO/IEC 29148:2011</i> .....	58
2.6.4	<i>Common Criteria for Information Technology Security Evaluation (CC)</i> .....	60
2.7	PROCESO: PMBoK.....	64
2.8	HERRAMIENTAS CASE: GESTIÓN DE REQUISITOS.....	64
2.8.1	<i>Evolución de herramientas de gestión de requisitos</i> .....	65
2.8.2	<i>Características generales de las herramientas de gestión de requisitos</i> .....	65
2.8.3	<i>La trazabilidad como parte de la gestión de requisitos</i> .....	66
2.8.4	<i>Herramientas en estudio</i> .....	67
2.8.5	<i>Características principales de las herramientas de gestión de requisitos</i> .....	68
2.9	INVESTIGACIONES RELACIONADAS AL REÚSO DE REQUISITOS .....	70
2.9.1	<i>Patrones de análisis (Fowler, 1997)</i> .....	70
2.9.2	<i>SIREN (Simple REuse of software requiremeNts)</i> .....	71
2.9.3	<i>Reutilización de Requisitos en el modelo Mecano</i> .....	72
2.9.4	<i>Patrones de Reutilización de Requisitos</i> .....	73
2.9.5	<i>Patrones de requisitos de software (Withall, 2010)</i> .....	74
2.9.6	<i>Técnicas de reúso: aplicación de dominios.</i> .....	75
2.9.7	<i>Reúso efectivo de modelos domóticos a través de requisitos genéricos</i> .....	76
2.9.8	<i>Evaluation of a systematic approach to requirements reuse</i> .....	77
2.9.9	<i>Otros estudios:</i> .....	77
<b>3</b>	<b>SOLUCIÓN PROPUESTA .....</b>	<b>81</b>
3.1	ASPECTOS GENERALES .....	81
3.2	MODELO DE REQUISITOS ORIENTADO AL REÚSO EFECTIVO (MORORE) .....	82
3.2.1	<i>Nivel de requisito individual</i> .....	86
3.2.2	<i>Nivel de patrón de requisitos</i> .....	97
3.2.3	<i>Nivel de estructura de tipos de requisitos</i> .....	102
3.3	MODELO DE CALIDAD .....	108
3.3.1	<i>Factores de calidad</i> .....	109
3.3.2	<i>Indicador de calidad</i> .....	113
3.3.3	<i>Activo de requisitos</i> .....	120
3.3.4	<i>Escalas de las métricas</i> .....	121
3.3.5	<i>Función de paso</i> .....	122
3.3.6	<i>Descripción de las relaciones del modelo de calidad</i> .....	124
3.3.7	<i>EJEMPLO: análisis de un patrón de requisitos</i> .....	148
3.4	MODELO DEL PROCESO DE REÚSO PARA MORORE .....	156
3.4.1	<i>Indexación</i> .....	158
3.4.2	<i>Recuperación</i> .....	171
3.4.3	<i>Aplicación de la metodología</i> .....	179

<b>4</b>	<b>EVALUACIÓN DE LA SOLUCIÓN .....</b>	<b>210</b>
4.1	EXPERIMENTACIÓN .....	210
4.1.1	<i>Metodología de Validación.....</i>	<i>210</i>
4.1.2	<i>Validación de Hipótesis .....</i>	<i>213</i>
<b>5</b>	<b>CONCLUSIONES, APORTACIONES Y TRABAJOS FUTUROS.....</b>	<b>223</b>
5.1	DISCUSIÓN Y CONCLUSIONES .....	223
5.2	APORTACIONES ORIGINALES .....	225
5.3	TRABAJOS FUTUROS .....	225
5.4	ACEPTACIÓN .....	226
5.4.1	<i>Publicaciones.....</i>	<i>226</i>
5.4.2	<i>Conferencias.....</i>	<i>229</i>
5.4.3	<i>Proyectos I+D .....</i>	<i>229</i>
5.4.4	<i>Proyectos fin de carrera .....</i>	<i>230</i>
5.4.5	<i>Aplicaciones.....</i>	<i>230</i>
5.4.6	<i>Universidades .....</i>	<i>231</i>
<b>6</b>	<b>REFERENCIAS.....</b>	<b>233</b>
<b>7</b>	<b>ANEXOS.....</b>	<b>245</b>
7.1	ANEXO 1: GRUPOS DE ALUMNOS CON CALIFICACIONES.....	245
7.1.1	<i>Data de alumnos del curso “Análisis y Diseño de Sistemas (ADS)” sede Piura.</i>	<i>245</i>
7.1.2	<i>Data de alumnos del curso “Análisis y Diseño de Sistemas (ADS)” sede Lima.</i>	<i>250</i>
7.1.3	<i>Data de alumnos del curso “Análisis y Diseño Orientado a Objetos (AYD)”</i>	
	<i>Escuela Tecnológica sede Piura. ....</i>	<i>252</i>
7.2	ANEXO 2: ENCUESTA DE MÉTRICAS DE REQUISITOS (PERSONAL).....	255
7.3	ANEXO 3: DETALLE DE LAS RESPUESTAS DE CADA ALUMNO .....	258
7.4	ANEXO 4: CONTENIDO DEL TRABAJO .....	315
7.5	ANEXO 5: ANÁLISIS ESTADÍSTICO .....	316
7.5.1	<i>Hipótesis de calidad .....</i>	<i>316</i>
7.5.2	<i>Hipótesis de tiempo.....</i>	<i>322</i>

# ÍNDICE DE TABLAS

TABLA 1 ESTUDIOS REALIZADOS POR STANDISH GROUP DESDE 1994 HASTA 2015: THE CHAOS REPORT .....	21
TABLA 2 ESTRUCTURA DE TIPOS DE REQUISITOS, ESTÁNDAR MÉTRICA V3.....	54
TABLA 3 ATRIBUTOS DE REQUISITOS, ESTÁNDAR MÉTRICA V3.....	55
TABLA 4 CUALIDADES DE REQUISITOS, ESTÁNDAR MÉTRICA V3.....	55
TABLA 5 ESTRUCTURA DE TIPOS DE REQUISITOS, ESTÁNDAR ESA .....	57
TABLA 6 ATRIBUTOS DE REQUISITOS, ESTÁNDAR ESA.....	58
TABLA 7 CUALIDADES DE REQUISITOS, ESTÁNDAR ESA.....	58
TABLA 8 ESTRUCTURA DE TIPOS DE REQUISITOS, ESTÁNDAR ISO/IEC/IEEE 29148.....	59
TABLA 9 ATRIBUTOS COMUNES DE REQUISITOS, ESTÁNDAR ISO/IEC/IEEE 29148:2011.....	59
TABLA 10 CUALIDADES DE REQUISITOS, ESTÁNDAR ISO/IEC/IEEE 29148:2011.....	60
TABLA 11 EJEMPLO DE LA NOMENCLATURA DE UN REQUISITO EN EL CC.....	62
TABLA 12 OTROS ESTUDIOS REALIZADOS RESPECTO AL REÚSO DE REQUISITOS.....	78
TABLA 13 EJEMPLO DE REQUISITO FUNCIONAL POTENCIALMENTE REUSABLE .....	84
TABLA 14 EJEMPLO DE REQUISITO NO FUNCIONAL POTENCIALMENTE REUSABLE .....	84
TABLA 15 EJEMPLO DE PATRÓN DE REQUISITOS.....	85
TABLA 16 EJEMPLO DE ESTRUCTURA DE TIPOS .....	85
TABLA 17 EJEMPLO DE FACTORES Y MÉTRICAS DE CALIDAD DE LOS NIVELES DE MORORE .....	86
TABLA 18 INSTANCIA DE LA CLASE REQUISITO .....	89
TABLA 19 INSTANCIA DE LA CLASE ATRIBUTO .....	90
TABLA 20 ATRIBUTOS PROPUESTOS PARA LA PLANTILLA CONFIGURABLE DE ESPECIFICACIÓN DE REQUISITOS .....	91
TABLA 21 INSTANCIA DE LA CLASE INTERMEDIA REQUISITO-ATRIBUTO.....	91
TABLA 22 INSTANCIA DE LA CLASE CONFIGURACIÓN.....	92
TABLA 23 INSTANCIA DE LA SUBCLASE AUTOMÁTICO.....	92
TABLA 24 INSTANCIA DE LA CLASE FINITO Y DE LA CLASE VALORFINITO .....	93
TABLA 25 INSTANCIA DE LA CLASE INFINITODEFINIDO.....	93
TABLA 26 INSTANCIA DE LA CLASE INFINITOINDEFINIDO .....	93
TABLA 27 INSTANCIA DE LA CLASE RELEVANCIA.....	94
TABLA 28 INSTANCIA DE LA CLASE TIPODATO .....	94
TABLA 29 INSTANCIA DE LA CLASE CONTROL.....	94
TABLA 30 INSTANCIA DE LA CLASE-ASOCIACIÓN ASOCIACIÓN .....	95
TABLA 31 INSTANCIA DE LA CLASE ELEMENTORELACIONADO .....	96
TABLA 32 INSTANCIA DE LA CLASE PARÁMETRO Y TIPODATO .....	97
TABLA 33 INSTANCIA DE LA CLASE PATRÓN .....	99
TABLA 34 INSTANCIA DE LA CLASE ATRIBUTO .....	100
TABLA 35 ATRIBUTOS PARA LA PLANTILLA DEL PATRÓN DE REQUISITOS DE MORORE .....	100
TABLA 36 INSTANCIA DE LA CLASE INTERMEDIA PATRÓN-ATRIBUTO .....	100
TABLA 37 INSTANCIA DE LA CLASE ELEMENTORELACIONADO .....	101
TABLA 38 INSTANCIA DE LA CLASE REQUISITOS.....	102
TABLA 39 INSTANCIA DE LA CLASE ESTRUCTURA .....	104
TABLA 40 INSTANCIA DE LA CLASE ATRIBUTO .....	105
TABLA 41 ATRIBUTOS PARA LA PLANTILLA DEL PATRÓN DE ESTRUCTURAS DE TIPOS.....	105
TABLA 42 INSTANCIA DE LA CLASE ESTRUCTURA-ATRIBUTO .....	105
TABLA 43 INSTANCIA DE LA CLASE ELEMENTORELACIONADO .....	106
TABLA 44 INSTANCIA DE LA CLASE TIPO .....	107
TABLA 45 INSTANCIA DE LA CLASE ASOCIACIÓN: ASOCIACIÓN.....	108
TABLA 46 CLASIFICACIÓN DE LOS INDICADORES DE CALIDAD .....	114
TABLA 47 EJEMPLO DEL PESO PONDERADO ASIGNADO A INDICADORES RESPECTO A UN FACTOR DE UN ACTIVO.....	120
TABLA 48 EJEMPLO DE LA APLICACIÓN DE LA ESCALA DE UN INDICADOR A LA ESCALA GENERAL.....	122
TABLA 49 RELACIÓN DE FACTORES E INDICADORES QUE PERMITEN EVALUAR LA CALIDAD DE LOS ACTIVOS .....	126
TABLA 50 PESOS DE LOS INDICADORES PARA EL FACTOR "ATOMICIDAD" DEL REQUISITO.....	127
TABLA 51 PESOS DE LOS INDICADORES PARA EL FACTOR "PRECISIÓN" DEL REQUISITO .....	127



TABLA 52 PESOS DE LOS INDICADORES PARA EL FACTOR "ABSTRACCIÓN" DEL REQUISITO .....	128
TABLA 53 PESOS DE LOS INDICADORES PARA EL FACTOR "INAMBIGÜEDAD" DEL REQUISITO.....	128
TABLA 54 PESOS DE LOS INDICADORES PARA FACTOR "COMPENSIBILIDAD" DEL REQUISITO .....	128
TABLA 55 PESOS DE LOS INDICADORES PARA EL FACTOR "MODIFICABILIDAD" DEL REQUISITO.....	128
TABLA 56 PESOS DE LOS INDICADORES PARA EL FACTOR "INTEGRIDAD" DEL REQUISITO.....	129
TABLA 57 PESOS DE LOS INDICADORES PARA EL FACTOR "SENCILLEZ" DEL REQUISITO.....	129
TABLA 58 PESOS DE LOS INDICADORES PARA EL FACTOR "ADAPTABILIDAD" DEL REQUISITO.....	129
TABLA 59 PESOS DE LOS INDICADORES PARA EL FACTOR "ATOMICIDAD" DEL PATRÓN DE REQUISITOS .....	130
TABLA 60 PESOS DE LOS INDICADORES PARA EL FACTOR "PRECISIÓN" DEL PATRÓN DE REQUISITOS.....	130
TABLA 61 PESOS DE LOS INDICADORES PARA EL FACTOR "ABSTRACCIÓN" DEL PATRÓN DE REQUISITOS.....	130
TABLA 62 PESOS DE LOS INDICADORES PARA EL FACTOR "INAMBIGÜEDAD" DEL PATRÓN DE REQUISITOS .....	130
TABLA 63 PESOS DE LOS INDICADORES PARA EL FACTOR "COMPENSIBILIDAD" DEL PATRÓN DE REQUISITOS .....	131
TABLA 64 PESOS DE LOS INDICADORES PARA EL FACTOR "MODIFICABILIDAD" DEL PATRÓN DE REQUISITOS .....	131
TABLA 65 PESOS DE LOS INDICADORES PARA EL FACTOR "INTEGRIDAD" DEL PATRÓN DE REQUISITOS .....	131
TABLA 66 PESOS DE LOS INDICADORES PARA EL FACTOR "SENCILLEZ" DEL PATRÓN DE REQUISITOS .....	131
TABLA 67 PESOS DE LOS INDICADORES PARA EL FACTOR "SIMPLICIDAD" DEL PATRÓN DE REQUISITOS.....	132
TABLA 68 PESOS DE LOS INDICADORES PARA EL FACTOR "ADAPTABILIDAD" DEL PATRÓN DE REQUISITOS .....	132
TABLA 69 PESOS DE LOS INDICADORES PARA EL FACTOR "INAMBIGÜEDAD" DEL PATRÓN DE ESTRUCTURAS .....	132
TABLA 70 PESOS DE LOS INDICADORES PARA EL FACTOR "COMPENSIBILIDAD" DEL PATRÓN DE ESTRUCTURAS .....	133
TABLA 71 PESOS DE LOS INDICADORES PARA EL FACTOR "MODIFICABILIDAD" DEL PATRÓN DE ESTRUCTURAS .....	133
TABLA 72 PESOS DE LOS INDICADORES PARA EL FACTOR "ORGANIZACIÓN" DEL PATRÓN DE ESTRUCTURAS.....	133
TABLA 73 PESOS DE LOS INDICADORES PARA EL FACTOR "SIMPLICIDAD" DEL PATRÓN DE ESTRUCTURAS.....	133
TABLA 74 PESOS DE LOS INDICADORES PARA EL FACTOR "CONSISTENCIA" DEL PATRÓN DE ESTRUCTURAS .....	134
TABLA 75 PESOS DE LOS INDICADORES PARA EL FACTOR "ADAPTABILIDAD" DEL PATRÓN DE ESTRUCTURAS .....	134
TABLA 76 NOTACIÓN PARA LA ESPECIFICACIÓN DE LOS RANGOS DE VALORES .....	135
TABLA 77 RANGOS DEL INDICADOR "NÚMERO DE PALABRAS" POR FACTOR DEL REQUISITO .....	135
TABLA 78 RANGOS DEL INDICADOR "PROMEDIO DE FRASES POR PÁRRAFO" POR FACTOR DEL REQUISITO .....	136
TABLA 79 RANGOS DEL INDICADOR "PROMEDIO DE PALABRAS POR ORACIÓN" POR FACTOR DEL REQUISITO .....	136
TABLA 80 RANGOS DEL INDICADOR "NÚMERO DE ABREVIATURAS Y ACRÓNIMOS" POR FACTOR DEL REQUISITO.....	136
TABLA 81 RANGOS DEL INDICADOR "PROMEDIO DE SIGNOS DE PUNTUACIÓN POR FRASE" POR FACTOR DEL REQUISITO .....	136
TABLA 82 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS CONECTIVOS" POR FACTOR DEL REQUISITO.....	137
TABLA 83 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS IMPRECISOS" POR FACTOR DEL REQUISITO .....	137
TABLA 84 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS DE DISEÑO" POR FACTOR DEL REQUISITO .....	137
TABLA 85 RANGOS DEL INDICADOR "NÚMERO DE ERRORES ORTOGRÁFICOS Y GRAMÁTICALES" POR FACTOR DEL REQUISITO.....	137
TABLA 86 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS AMBIGUOS" POR FACTOR DEL REQUISITO.....	138
TABLA 87 RANGOS DEL INDICADOR "NÚMERO DE FORMAS VERBALES IMPERATIVAS E INDICATIVAS " POR FACTOR DEL REQUISITO .....	138
TABLA 88 RANGOS DEL INDICADOR "NÚMERO DE FORMAS VERBALES CONDICIONALES Y COMPUESTAS" POR FACTOR DEL REQUISITO .....	138
TABLA 89 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS DE DOMINIO" POR FACTOR DEL REQUISITO.....	138
TABLA 90 RANGOS DEL INDICADOR "NÚMERO DE RELACIONES CON OTROS ELEMENTOS" POR FACTOR DEL REQUISITO .....	139
TABLA 91 RANGOS DEL INDICADOR "NÚMERO DE SOLAPAMIENTOS" POR FACTOR DEL REQUISITO.....	139
TABLA 92 RANGOS DEL INDICADOR "NÚMERO DE VERSIONES" POR FACTOR DEL REQUISITO.....	139
TABLA 93 RANGOS DEL INDICADOR "NÚMERO DE ÉXITOS" POR FACTOR DEL REQUISITO .....	139
TABLA 94 RANGOS DEL INDICADOR "NÚMERO DE PALABRAS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	140
TABLA 95 RANGOS DEL INDICADOR "NÚMERO DE ELEMENTOS DE LA SOLUCIÓN" POR FACTOR DEL PATRÓN DE REQUISITOS .....	140
TABLA 96 RANGOS DEL INDICADOR "PROMEDIO DE FRASES POR PÁRRAFO" POR FACTOR DEL PATRÓN DE REQUISITOS.....	140
TABLA 97 RANGOS DEL INDICADOR "PROMEDIO DE PALABRAS POR ORACIÓN" POR FACTOR DEL PATRÓN DE REQUISITOS...	140
TABLA 98 RANGOS DEL INDICADOR "NÚMERO DE ABREVIATURAS Y ACRÓNIMOS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	141
TABLA 99 RANGOS DEL INDICADOR "PROMEDIO DE SIGNOS DE PUNTUACIÓN POR FRASE" POR FACTOR DEL PATRÓN DE REQUISITOS .....	141
TABLA 100 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS CONECTIVOS" POR FACTOR DEL PATRÓN DE REQUISITOS....	141

TABLA 101 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS IMPRECISOS" POR FACTOR DEL PATRÓN DE REQUISITOS.....	141
TABLA 102 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS DE DISEÑO" POR FACTOR DEL PATRÓN DE REQUISITOS .....	142
TABLA 103 RANGOS DEL INDICADOR "NÚMERO DE ERRORES ORTOGRÁFICOS Y GRAMATICALES" POR FACTOR DEL PATRÓN DE REQUISITOS .....	142
TABLA 104 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS AMBIGUOS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	142
TABLA 105 RANGOS DEL INDICADOR "NÚMERO DE FORMAS VERBALES IMPERATIVAS E INDICATIVAS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	142
TABLA 106 RANGOS DEL INDICADOR "NÚMERO DE FORMAS VERBALES CONDICIONALES Y COMPUESTAS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	143
TABLA 107 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS DE DOMINIO" POR FACTOR DEL PATRÓN DE REQUISITOS.....	143
TABLA 108 RANGOS DEL INDICADOR "NÚMERO DE RELACIONES CON OTROS ELEMENTOS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	143
TABLA 109 RANGOS DEL INDICADOR "NÚMERO DE DEPENDENCIAS ENTRE ELEMENTOS DEL ACTIVO" POR FACTOR DEL PATRÓN DE REQUISITOS .....	144
TABLA 110 RANGOS DEL INDICADOR "NÚMERO DE VERSIONES" POR FACTOR DEL PATRÓN DE REQUISITOS.....	144
TABLA 111 RANGOS DEL INDICADOR "NÚMERO DE ÉXITOS" POR FACTOR DEL PATRÓN DE REQUISITOS .....	144
TABLA 112 RANGOS DEL INDICADOR "NÚMERO DE ELEMENTOS DE LA SOLUCIÓN" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	144
TABLA 113 RANGOS DEL INDICADOR "PROMEDIO DE FRASES POR PÁRRAFO" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .	145
TABLA 114 RANGOS DEL INDICADOR "PROMEDIO DE SÍLABAS POR PALABRA" POR FACTOR DEL PATRÓN DE ESTRUCTURAS.	145
TABLA 115 RANGOS DEL INDICADOR "NÚMERO DE ABREVIATURAS Y ACRÓNIMOS" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	145
TABLA 116 RANGOS DEL INDICADOR "NÚMERO DE SIGNOS DE PUNTUACIÓN POR FRASE" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	145
TABLA 117 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS CONECTIVOS" POR FACTOR DEL PATRÓN DE ESTRUCTURAS.	146
TABLA 118 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS IMPRECISOS" POR FACTOR DEL PATRÓN DE ESTRUCTURAS..	146
TABLA 119 RANGOS DEL INDICADOR "NÚMERO DE ERRORES ORTOGRÁFICOS Y GRAMATICALES" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	146
TABLA 120 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS AMBIGUOS" POR FACTOR DEL PATRÓN DE ESTRUCTURAS...	146
TABLA 121 RANGOS DEL INDICADOR "NÚMERO DE TÉRMINOS DE DOMINIO" POR FACTOR DEL PATRÓN DE ESTRUCTURAS.	147
TABLA 122 RANGOS DEL INDICADOR "NÚMERO DE RELACIONES CON OTROS ELEMENTOS" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	147
TABLA 123 RANGOS DEL INDICADOR "NÚMERO DE DEPENDENCIAS ENTRE ELEMENTOS DEL ACTIVO" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	147
TABLA 124 RANGOS DEL INDICADOR "NÚMERO DE NIVELES DE ANIDAMIENTO" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	148
TABLA 125 RANGOS DEL INDICADOR "NÚMERO DE ÉXITOS" POR FACTOR DEL PATRÓN DE ESTRUCTURAS .....	148
TABLA 126 EJEMPLO DE PATRÓN DE REQUISITOS MEJORABLE. ....	150
TABLA 127 RANGO DE VALORES DEL INDICADOR "NÚMERO DE TÉRMINOS CONECTIVOS" RESPECTO AL FACTOR "COMPENSIBILIDAD" DE UN PATRÓN DE REQUISITOS (OBTENIDO DE LA TABLA 100).....	151
TABLA 128 RANGO DE VALORES DEL INDICADOR "NÚMERO DE TÉRMINOS AMBIGUOS" RESPECTO AL FACTOR "COMPENSIBILIDAD" DE UN PATRÓN DE REQUISITOS (OBTENIDO DE LA TABLA 104).....	151
TABLA 129 RANGO DE VALORES DEL INDICADOR "NÚMERO DE TÉRMINOS CONECTIVOS" RESPECTO AL FACTOR "INTEGRIDAD" DE UN PATRÓN DE REQUISITOS (OBTENIDO DE LA TABLA 100) .....	152
TABLA 130 RESULTADOS DEL ANÁLISIS DEL PATRÓN DE REQUISITOS "GESTIÓN DE INVENTARIOS" .....	155
TABLA 131 EJEMPLO DE PATRÓN DE REQUISITOS MEJORADO .....	156
TABLA 132. DICCIONARIO DEL ACTIVO SOFTWARE .....	167
TABLA 133. REQUISITOS DEL SIIU.....	181
TABLA 134. PATRÓN DE REQUISITOS PARA UNA SITUACIÓN ESPECÍFICA.....	185
TABLA 135. RELACIÓN ENTRE LOS FACTORES DE CALIDAD Y LOS INDICADORES DE CALIDAD (OBTENIDA DE LA TABLA 49)....	186
TABLA 136. RESULTADOS DE CADA INDICADOR. ....	187
TABLA 137. RESULTADO DE CADA FACTOR. ....	188
TABLA 138. ANÁLISIS Y CLASIFICACIÓN DE LA INFORMACIÓN DEL PATRÓN. ....	190

TABLA 139. DICCIONARIO DEL PATRÓN DE REQUISITOS. ....	191
TABLA 140. PATRÓN DE REQUISITOS. ....	193
<i>TABLA 141. RESULTADOS DE LA BÚSQUEDA. ....</i>	<i>197</i>
TABLA 142. EL PATRÓN DE REQUISITOS (OBTENIDO DE LA TABLA 140) .....	200
TABLA 143. PLANTILLA DE REQUISITOS DEL PROYECTO DE SOFTWARE PARA EL GRUPO CS. ....	201
TABLA 144. REQUISITO R01 INTEGRADO .....	201
TABLA 145. REQUISITO R02 INTEGRADO .....	201
TABLA 146. REQUISITO R03 INTEGRADO .....	202
TABLA 147. REQUISITO R04 INTEGRADO .....	202
TABLA 148. REQUISITO R05 INTEGRADO .....	202
TABLA 149. REQUISITO R06 INTEGRADO .....	203
TABLA 150. REQUISITO R04 INSTANCIADO.....	204
TABLA 151. REQUISITO R05 INSTANCIADO.....	205
TABLA 152. REQUISITO RA001 .....	206
TABLA 153. REQUISITO RA002 .....	206
TABLA 154. REQUISITO RA003 .....	206
TABLA 155. REQUISITO RA004 .....	207
TABLA 156. REQUISITO RA005 .....	207
TABLA 157 HERRAMIENTAS USADAS EN EL EXPERIMENTO .....	211
TABLA 158 DISTRIBUCIÓN DE LOS ALUMNOS POR HERRAMIENTA.....	212
TABLA 159 CRITERIOS DE EVALUACIÓN PARA LA PRESENTACIÓN ESCRITA. ....	214
TABLA 160 CRITERIOS DE EVALUACIÓN PARA LA PRESENTACIÓN ORAL .....	214
TABLA 161 SUMA DE LOS PORCENTAJES DE LOS CRITERIOS DE EVALUACIÓN .....	214
TABLA 162 RESUMEN DEL ANÁLISIS ESTADÍSTICO DE LAS VARIABLES RESPECTO AL TIEMPO .....	220

# ÍNDICE DE FIGURAS

FIGURA 1 PROCESO METODOLÓGICO DEL DESARROLLO DE LA TESIS. ....	29
FIGURA 2 PASOS PARA LA DEFINICIÓN DE MORORE. ....	32
FIGURA 3 PROCESO DE LA INGENIERÍA DE REQUISITOS. ....	41
FIGURA 4 ESQUEMA DE PROCESO DE RECUPERACIÓN DE INFORMACIÓN. FUENTE: (SECO, 2009) .....	45
FIGURA 5 ORGANIZACIÓN DE REQUISITOS DE SEGURIDAD, ESTÁNDAR CC. ....	62
FIGURA 6 USO DE REQUISITOS DE SEGURIDAD, ESTÁNDAR CC. ....	63
FIGURA 7 EJEMPLO DE MODELO DE UN PATRÓN DE ANÁLISIS SEGÚN FOWLER.....	71
FIGURA 8 PATRONES DE REQUISITOS SOFTWARE DE WITHALL. ....	75
FIGURA 9 ESQUEMA GENERAL DEL MODELO DE REQUISITOS ORIENTADO AL REÚSO EFECTIVO (MORORE).....	83
FIGURA 10 MODELO DEL NIVEL DE REQUISITO INDIVIDUAL DE MORORE .....	87
FIGURA 11 MODELO DEL NIVEL DE PATRÓN DE REQUISITOS DE MORORE .....	98
FIGURA 12 MODELO DEL NIVEL DE PATRÓN DE ESTRUCTURA DE TIPOS DE REQUISITOS DE MORORE.....	103
FIGURA 13 MODELO DE CALIDAD DE ACTIVOS DE REQUISITOS DE MORORE .....	109
FIGURA 14 FUNCIONES DE PASO DE UN INDICADOR RESPECTO A UN FACTOR DE UN TIPO DE ACTIVO.....	124
FIGURA 15. PROCESO DE INDEXACIÓN Y DE RECUPERACIÓN PROPUESTO. ....	157
FIGURA 16. ETAPA DE IDENTIFICACIÓN DEL ACTIVO.....	160
FIGURA 17. ETAPA DE CONTROL DE CALIDAD.....	163
FIGURA 18. ETAPA DE ADAPTACIÓN. ....	165
FIGURA 19. ETAPA DE TRANSFORMACIÓN. ....	168
FIGURA 20. ETAPA DE ALMACENAMIENTO. ....	170
FIGURA 21. ETAPA DE CONSULTA. ....	172
FIGURA 22. ETAPA DE SELECCIÓN. ....	174
FIGURA 23. ETAPA DE INTEGRACIÓN.....	176
FIGURA 24. ETAPA DE ADECUACIÓN. ....	177
FIGURA 25. DIAGRAMA CONCEPTUAL DEL PATRÓN. ....	182
FIGURA 26. ESTRUCTURA ORGANIZACIONAL DEL GRUPO CS.....	195
FIGURA 27. ESTRUCTURA DE LAS EMPRESAS DEL GRUPO CS .....	196
FIGURA 28 INTERVALOS DE CONFIANZA AL 95% - MÉTODO DE BONFERRONI .....	215
FIGURA 29 GRÁFICOS DE RESIDUOS FRENTE A VALORES PREVISTOS CORRECTOS E INCORRECTOS (SÁNCHEZ, 2017).....	321

## DICCIONARIO DE DATOS

- Activo software: llamado también artefacto software. es un entregable del proceso de desarrollo de software.
- Nivel de activo de MORORE: es el tipo de elemento reusable definido en el Modelo propuesto en la tesis. Son 3 tipos de activos definidos: requisito individual, patrón de requisitos y patrón de estructura de tipos de requisitos.
- Patrón de estructura de tipos de requisitos: conjunto de clasificaciones de requisitos típicamente jerárquicas que permiten organizar a los requisitos para su gestión en un proyecto específico.
- Patrón de requisitos: conjunto de requisitos relacionados que enfocan una solución a un problema de la ingeniería de requisitos dentro de un proyecto de software.

# INTRODUCCIÓN

## 1. Introducción

### 1.1 Contexto de la investigación

La investigación, descrita en la presente tesis, se desarrolla dentro del grupo de investigación Knowledge Reuse Group (KR) (KR, 2017) de la Universidad Carlos III de Madrid (UC3M) y la Universidad de Piura (UDEP). El grupo KR centra su trabajo en las áreas de interés de Representación y Recuperación del Conocimiento y la Recuperación y Reutilización aplicada a la Ingeniería del Software (KR, 2017). En concordancia con el grupo KR, el Modelo de Requisitos Orientado al Reúso Efectivo (MORORE) se enmarca en la línea de investigación de reúso de activos software, orientado en los dos ámbitos: representación y recuperación del conocimiento aplicada a la ingeniería de requisitos.

Como mencionamos, orientamos este trabajo de investigación a la ingeniería de requisitos (IR), una rama de la ingeniería de software. Esta orientación se debe a que la IR se constituye como un elemento esencial del proceso de desarrollo de software. Muchos estudios, descritos en el acápite 1.2 (Descripción del problema), demuestran que la mayor cantidad de los problemas en el desarrollo de software se originan en una mala especificación y gestión de requisitos. Es por ello que surgen muchos estándares, herramientas, métodos y técnicas con la intención de dar solución al problema.

Nuestro trabajo de investigación aporta una solución al problema de la ingeniería de requisitos: el empleo de técnicas de reúso aplicadas a activos software del ámbito de la ingeniería de requisitos, con el fin de mejorar la calidad de la especificación de los requisitos. El trabajo comienza con la obtención de los modelos de representación y recuperación abstraídos de los libros, estándares, artículos y otras fuentes referentes al tema de la ingeniería requisitos. Estos modelos ayudaron a proponer un modelo de requisitos general (MORORE) que soporte la representación de los diferentes modelos estudiados y otros posibles.

En este sentido, el desarrollo de la presente investigación concierne a los temas de interés descritos a continuación:

- Ingeniería de software. De forma general coincidimos con Sommerville (Sommerville, 2016) en que la ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta su mantenimiento. En consecuencia, la ingeniería de software se constituye en el tema general de interés, el cual abarca el ámbito del trabajo de investigación en curso: la ingeniería de requisitos.
- Ingeniería de requisitos. Definimos a la ingeniería de requisitos como la rama de la ingeniería de software que se encarga de la captación, análisis, validación y verificación de la información relativa a las necesidades de los usuarios; de la invención de la especificación correcta y completa de los requisitos propiamente dichos, dentro de un proceso evolutivo y de negociación; y del mantenimiento o gestión de los requisitos dentro de todo el proceso de desarrollo de software. Esta tesis doctoral propone una forma de mejorar el proceso de la ingeniería de requisitos aplicando el reúso de requisitos. Por tanto, la ingeniería de requisitos está directamente relacionada con la investigación que desarrollamos.

# INTRODUCCIÓN

---

- Patrones. La idea de patrón propone la forma de transmitir experiencias de soluciones exitosas a problemas recurrentes (Gamma, et al., 1995). La solución debe ser reutilizable, para problemas similares en distintas circunstancias. Esta idea surgida inicialmente en el ámbito de la Arquitectura Civil, es trasladada al campo de la informática para transmitir experiencias exitosas en el diseño de sistemas informáticos. Nuestra investigación propone el traslado de la idea de los patrones al ámbito de la ingeniería de requisitos.
- Herramientas CASE (Computer Aided Software Engineering): Gestión de requisitos. CASE (Sommerville, 2016), es el nombre que se le da al software que se utiliza para ayudar a las actividades del proceso de desarrollo de software: la planificación, la ingeniería de requisitos, el diseño, el desarrollo de programas, las pruebas y el mantenimiento. En consecuencia, una herramienta CASE permitirá a los desarrolladores de software acelerar el proceso para el que han sido diseñadas. Para el caso de una herramienta CASE orientada a la gestión de requisitos, permitirá automatizar o apoyar una o más actividades de la ingeniería de requisitos: captura de necesidades, y especificación y gestión de requisitos. La investigación en curso propone que MORORE sea soportado por una herramienta de gestión de requisitos, motivo por el cual está relacionada la investigación con este tipo de aplicaciones.
- Calidad. Grado en el que un conjunto de características inherentes a un objeto (producto, servicio, proceso, persona, organización, sistema o recurso) cumple con los requisitos (ISO-9000, 2017)). Este tema se relaciona con esta tesis doctoral, pues en el modelo proponemos una forma de medir la calidad de los activos de la ingeniería de requisitos. Esta medición permitirá almacenar los activos reutilizables de MORORE con un mínimo de calidad para su reúso efectivo en un nuevo proyecto.
- Recuperación de la información. Parte de la ciencia de la computación que estudia la recuperación de información (no de meros datos) de una colección de documentos (Baeza-Yates & Ribeiro-Neto, 1999). La recuperación de documentos tiene como objetivo satisfacer la necesidad de información de un usuario, normalmente expresada en lenguaje natural. La presente investigación se relaciona con este tema, pues propone de modo complementario para el uso del modelo, un proceso de indización y recuperación de los productos reutilizables propuestos en MORORE.

## 1.2 Descripción del problema

Al incrementarse el uso y la complejidad de los sistemas de información en la década de los 60, aparecen los problemas que Pressman describe como una “aflicción crónica” del software<sup>1</sup> (Pressman, 2002). Esta aflicción se traduce en muchos y serios inconvenientes del producto y del desarrollo del software.

En este contexto surge la ingeniería de software como una disciplina de la ingeniería que comprende todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema hasta el mantenimiento durante su explotación (Sommerville, 2016). La ingeniería software proporciona muchos estándares,

---

<sup>1</sup> Muchos científicos del área han denominado como “crisis” a los problemas asociados con el desarrollo de software. Pressman sugiere que el término adecuado para definir los problemas del proceso de desarrollo de software en su proceso evolutivo es “aflicción crónica” (Pressman, 2002).



# INTRODUCCIÓN

herramientas y técnicas para cubrir todos los aspectos técnicos y actividades de gestión de un proyecto de software. El fin de esta disciplina es construir eficientemente un software de calidad.

En la actualidad la ingeniería de software ha logrado un gran avance tecnológico; a pesar de ello, muchos estudios indican que existen importantes deficiencias en diferentes ámbitos que impiden que muchos proyectos de software obtengan resultados aceptables. Por ejemplo, en la Tabla 1 se muestran los resultados generales de los estudios realizados por The Standish Group International (Standish, 2016). Los estudios fueron realizados desde 1994 hasta su informe en 2015 en más 50000 proyectos de tecnologías de la información alrededor del mundo. Los resultados determinan que desde 1994 hasta 2015 el porcentaje de proyectos exitosos ha aumentado en un 81%, mientras que el porcentaje de los proyectos completos pero deficientes ha disminuido en un 2% y el porcentaje de proyectos cancelados se ha reducido a un 39% aproximadamente.

PROYECTO	1994	1998	2000	2002	2004	2006	2008	2010	2012	2014	2015
Exitoso	16%	26%	28%	34%	29%	35%	32%	37%	39%	28%	29%
Completo pero deficiente	53%	46%	49%	51%	53%	46%	44%	42%	43%	55%	52%
Cancelado	31%	28%	23%	15%	18%	19%	24%	21%	18%	17%	19%

*Tabla 1 Estudios realizados por Standish Group desde 1994 hasta 2015: The Chaos Report*

Podríamos decir que se ha producido una notable mejora, pero lamentablemente no es suficiente. Aún el porcentaje de proyectos exitosos es bajo, y el porcentaje de los proyectos completos pero deficientes y de los proyectos cancelados es alto. Todo esto se traduce en pérdidas millonarias en el mundo del proceso de desarrollo de software. En consecuencia, podemos decir que falta mucho por mejorar.

En la misma línea, otros estudios relacionados con la ingeniería de software señalan cuáles son los factores principales del fracaso en los proyectos de desarrollo de software. Por ejemplo Milagros Ibáñez y Helmut Rempp en su informe European User Survey Analysis (Ibáñez & Rempp, 1996), muestran que la especificación y gestión de requisitos son los factores principales de los problemas en la producción de software. Otros estudios indican que más del 50% de los encuestados asume como “problema mayor” la mala especificación y gestión de requisitos ( (Shah & Patel, 2014), (Manoel, et al., 2013), (Pressman & Maxim, 2015), (Sommerville, 2016), (Braude & Bernstein, 2016)) .

Del mismo modo Frederick Brooks (Brooks, 1987) dice: “lo más difícil al construir un sistema software es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual del desarrollo de software es tan difícil como el establecimiento de los requisitos técnicos detallados.” Y luego agrega: “ninguna otra parte del trabajo afecta tanto al software como la mala definición de los requisitos, pues es la más difícil de corregir más adelante... por tanto, la tarea más importante que el ingeniero de software hace para el cliente es la extracción y el refinamiento iterativos de los requisitos del producto.”

# INTRODUCCIÓN

---

Por su parte Eric Braude (Braude & Bernstein, 2016) dice que la mayor parte de los defectos encontrados en el software entregado se originan durante el análisis de los requisitos. En general, estos defectos también son los más caros de reparar. Se estima que reparar los defectos originados en la fase de requisitos es entre 20 y 50 veces más costoso si se permite que se propaguen al resto del proceso de desarrollo.

Adicionalmente, Christel y Kang (Christel & Kang, 1992) identifican algunos inconvenientes que hacen difícil el trabajo con los requisitos:

- Problemas de alcance: la especificación del ámbito funcional que abarcan los requisitos es una tarea complicada. Típicamente los límites del sistema no se definen bien, tampoco los detalles técnicos. Esto origina una confusión en los objetivos generales.
- Problemas de comprensión: los clientes y usuarios muchas veces no están completamente seguros de lo que necesitan. No existe un total entendimiento del problema a resolver con el sistema.
- Problemas de volatilidad: los requisitos cambian con el tiempo; los requisitos evolucionan durante el proceso mismo de desarrollo de software. Los ingenieros deben hacer una labor organizada para afrontar este entorno cambiante.

Marrero (Marrero, et al., 2008), agrega respecto a los problemas en la ingeniería de requisitos: “el problema es que generalmente los requisitos son redactados en lenguaje natural, cuya falta de normalización y ambigüedad hace que sean difíciles de representar, procesar y, por supuesto, recuperar y reutilizar”. Asimismo, Parra también considera que uno de los factores que influyen en la calidad de requisitos es la composición sintáctica y semántica de los requisitos escritos en lenguaje natural. En este sentido se convierte en una necesidad que los requisitos tengan un mínimo de calidad para que puedan ser reusados en un nuevo proyecto.

De igual forma, otros estudios también coinciden que la mayor cantidad de errores en los proyectos de desarrollo de ingeniería son originados la etapa de obtención, análisis, especificación y gestión de requisitos. Y cuyos errores son los más costosos de corregir si son detectados en las etapas finales de los proyectos ( INCOSE, 2017), (SCRIBD, 2017), (Govindaraju & Wiradanti, 2015), (Laplante, 2013), (Horkoff & Yu, 2016)).

De acuerdo con los estudios mencionados, sostenemos que el mundo científico y tecnológico debe concentrar, en mayor medida, sus esfuerzos para la mejora de las actividades del trabajo con requisitos. Son muchas las dificultades que afrontan los desarrolladores de software al capturar las necesidades y especificar los requisitos. Para capturar las necesidades se requiere una interacción cuidadosa con los interesados en la aplicación. Asimismo, para especificar los requisitos se debe realizar un estudio minucioso y metódico de la información adquirida. Este estudio debe conducir a una verdadera comprensión de las necesidades, con la intención de definir correctamente los requisitos y poder expresarlos de forma concreta y detallada (Génova, et al., 2013) .

## 1.3 Importancia del problema

En concordancia con los estudios expresados en el apartado 1.2 (Descripción del problema), asumimos la postura que en la fase de análisis de software, específicamente en la especificación y gestión de requisitos, se producen los principales problemas que en la actualidad afronta el proceso de desarrollo de software (SDP: *Software Development Process*).

Por tanto, podemos mencionar que todos los problemas relacionados con el trabajo con requisitos pueden producir, en un plazo no determinado, perjuicios no sólo económicos sino también daños personales (incluso pérdida de vidas humanas), daños ambientales de diversa gravedad, problemas sociales, etc. En consecuencia, la fase de análisis de software se ha convertido en un tema crucial. Pero a pesar de la relevancia del tema, las actividades de esta fase todavía están sujetas a procesos de producción informal, parciales y en algunos casos no confiables por parte de las organizaciones dedicadas al desarrollo de software (Génova, et al., 2017).

En consecuencia, actualmente muchos científicos orientan sus esfuerzos a la investigación para el mejoramiento de la fase de análisis de software. Tales son los estudios relacionados a esta fase que se ha convertido en una disciplina en sí misma, la ingeniería de requisitos.

La ingeniería de requisitos es la disciplina que se encarga de la captura de necesidades, y especificación y gestión de requisitos de un proyecto de software. En tal sentido, y por lo mencionado el apartado 1.2, la ingeniería de requisitos ejerce un papel primordial en el proceso de desarrollo de software: definir qué se desea producir.

Realizar de forma adecuada el proceso de la ingeniería de requisitos disminuye significativamente la probabilidad de fracaso de un proyecto de desarrollo de software. Los requisitos bien definidos permiten conocer específicamente qué debe ser capaz de realizar el software por desarrollar. En consecuencia, los ingenieros tendrán mejores opciones de éxito para implementar el software que contenga las capacidades especificadas descritas en los requisitos.

La ingeniería de requisitos permite la gestión adecuada de los requisitos de un proyecto de desarrollo de software. Cada actividad de la ingeniería de requisitos consiste de una serie de pasos organizados y bien definidos que conllevan a la especificación correcta y completa de los requisitos. Estos requisitos vienen a ser las metas que debe alcanzar el futuro software. En consecuencia, todas las actividades del proyecto de desarrollo de software deben estar orientadas a conseguir de forma eficiente las metas en mención.

La ingeniería de requisitos bien desarrollada mejora la capacidad para realizar planificaciones de los procesos de proyectos de desarrollo de software. El conocer específicamente qué se tiene que desarrollar permite la proyección efectiva de las actividades, recursos, costos, tiempos, subproductos, controles, etc. del proyecto de desarrollo de software.

# INTRODUCCIÓN

---

La ingeniería de requisitos efectiva y eficiente permite la disminución de los costos y retrasos del proyecto. Los requisitos bien definidos conllevan a orientar las actividades, recursos y esfuerzos de manera eficiente para el alcance de los objetivos. Muchos estudios han demostrado que reparar errores por un mal desarrollo, si no es descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la fase de ingeniería de requisitos (Braude & Bernstein, 2016).

Un buen proceso de la ingeniería de requisitos mejora la calidad del software. Los requisitos bien especificados permitirán probar efectivamente el software. La calidad en el software tiene que ver con el cumplimiento de un conjunto de requisitos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).

La ingeniería de requisitos bien desarrollada mejora la comunicación entre los involucrados en el proceso de desarrollo de software. La especificación de requisitos representa una forma de consenso entre clientes y el equipo de desarrollo. Si el consenso para la especificación de los requisitos entre clientes e ingenieros no ocurre, el proyecto está destinado a no tener éxito.

Los resultados de una eficiente y efectiva ingeniería de requisitos permiten evaluar la productividad. Tener claro el objetivo y los requisitos del proyecto de desarrollo de software permitirá a los ingenieros conocer cuándo se ha llegado a la meta y por tanto medir la productividad del equipo de desarrollo.

Todo lo mencionado en este apartado expresa la importancia de realizar un proceso de la ingeniería de requisitos efectivo y eficiente para un proyecto de desarrollo de software. Hablamos de eficiencia de la ingeniería de requisitos, cuando nos referimos al mejoramiento del tiempo y por lo tanto de recursos en el proceso de desarrollo de requisitos para un proyecto de software. Y referimos la efectividad de la ingeniería de requisitos, como el mejoramiento de la calidad de los requisitos producidos en un proyecto de software.

La presente tesis doctoral propone un modelo de 3 niveles que permite representar para su almacenamiento los activos respectivos de la ingeniería de requisitos. Y luego, poder recuperar, en forma sencilla, estos activos para ser usados en un nuevo proyecto. En otras palabras, proponemos con MORORE, un desarrollo eficiente de la ingeniería de requisitos.

Asimismo, con MORORE, planteamos un modelo de calidad que permita medir la calidad de los activos de la ingeniería de requisitos que se desean almacenar. Este modelo consta de un conjunto de factores y métricas de calidad que permita el control de calidad de los activos potencialmente reusables para que puedan ser almacenados. Esto garantizaría un mínimo aceptable de calidad de los activos que se reúsen en un nuevo proyecto de software.

Podemos mencionar como los dos puntos característicos de MORORE: la representación de activos para su fácil reúso y el control de calidad de estos activos cuando son identificados o creados. Estos dos puntos permitirían, un proceso con reúso de la ingeniería de requisitos eficiente y efectivo respectivamente.

## 1.4 Hipótesis de investigación

La hipótesis que se plantea validar en la presente tesis doctoral es:

Si se dispone de activos de la ingeniería de requisitos de calidad (evaluados por un modelo de calidad compuesto de factores y métricas específicas), creados bajo un modelo de representación general que facilite el reúso con un soporte automático.

ENTONCES se podrá:

- *Reducir el tiempo* en el proceso de especificación de requisitos de un proyecto. El reúso de software, como hemos mencionado, se entiende como el uso en un nuevo proyecto de un activo software creado anteriormente y con un mínimo de cambios. Si tenemos elementos ya creados que podemos utilizar en el desarrollo de un nuevo proyecto, redundará en un ahorro de tiempo y costo del proceso.
- *Garantizar un mínimo de calidad* en la especificación de requisitos de un proyecto. MORORE propone la aplicación factores y métricas para el control de calidad en los activos software antes de su respectivo almacenamiento. En este sentido, los activos reusables almacenados a partir de MORORE tendrían un mínimo de calidad al momento del reúso de los mismos para un nuevo proyecto.

En consecuencia, planteamos un eficiente y efectivo reúso de activos de requisitos para mejorar el proceso de la ingeniería de requisitos. Y, por la importancia de la ingeniería de requisitos en el desarrollo de software, podemos decir que también es un aporte para mejorar todo el proceso de desarrollo de software.

## 1.5 Objetivos

El **objetivo general** de la presente tesis doctoral es:

Desarrollar un modelo de representación general para tres niveles de activos de requisitos y un modelo de calidad compuesto de factores y métricas específicas, que permita el reúso eficiente y efectivo respectivamente de requisitos, con una guía de pasos para el almacenamiento y recuperación de los activos.

Este modelo define una propuesta enmarcada en el empleo de técnicas de reúso orientada al ámbito de la ingeniería de requisitos, que conlleve a la mejora de la especificación de requisitos.

# INTRODUCCIÓN

---

A partir del objetivo mencionado, podemos definir los objetivos específicos siguientes:

- Objetivo específico 1:

*Definir la representación tres niveles de activos software que permita el reúso eficiente de activos de la ingeniería de requisitos. Estos niveles son los seleccionados para alcanzar los fines generales de la presente tesis: Representación individual de requisitos, Representación de conjuntos de requisitos (patrón de requisitos), Representación de estructuras de tipos de requisitos (patrón de estructuras de tipos). Esta representación está orientada a facilitar el reúso haciéndolo eficiente.*

- Objetivo específico 2:

*Definir un modelo de calidad con factores y métricas específicas que permita el reúso efectivo de activos de la ingeniería de requisitos. Estos factores y métricas están definidos para cada uno de los tres niveles de activos software seleccionados. Los factores y métricas de calidad se orientan a garantizar un mínimo de calidad de los activos reusables. Complementariamente se define una guía de pasos para el almacenamiento y recuperación de los activos.*

- Objetivo específico 3:

*Crear un conjunto de activos concretos, específicamente patrones de requisitos, que permitan el reúso eficiente y efectivo de los requisitos y, de esta manera, demostrar la validez de la propuesta de esta tesis doctoral.*

## 1.6 Aproximación a la solución

Como ya mencionamos, en la actualidad existen muchas propuestas de solución para mejorar la ingeniería de requisitos. Tenemos, por ejemplo:

- La creación de muchos estándares ( (INCOSE, 2017), (IEEE, 1990), (ESA, 2017), (ISO/IEC/IEEE-29148, 2017), (Metrica-3, 2017), (NTP, 2017), (CC, 2017), (SWEBOK, 2017)) relacionados con la ingeniería de requisitos que sirven como marco de referencia a los ingenieros para el desarrollo efectivo y eficiente de sus proyectos.
- El desarrollo de herramientas automáticas para la especificación y gestión de requisitos ( (CORE, 2017), (DOORS, 2017), (IRQA, 2017), (RDD-100, 2017),

# INTRODUCCIÓN

---

SIREN (Nicolás, et al., 2003), (Reppro, 2017), (SERENA, 2017), swREUSER (swReuser, 2017), REMM-Studio (Moros, et al., 2008)). Estas herramientas permiten a los ingenieros optimizar los procesos de desarrollo y gestión de los requisitos dentro de un proyecto específico.

- La creación de muchos métodos y técnicas que facilitan de diferentes formas el proceso de la ingeniería de requisitos. Encontramos trabajos cuya finalidad es formalizar y mejorar la especificación de requisitos mediante plantillas de requisitos ( (Abushark, et al., 2016), (Ito, 2016), (Srivastava, 2013), (ISO/IEC/IEEE-29148, 2017)).
- Propuestas de trabajos relacionado con la medición y control de calidad en la especificación de requisitos en lenguaje natural ((Génova et al. 2013), (Fuentes, et al., 2016), (Saavedra, et al., 2013), (Zapata, et al., 2006)). Estos trabajos tienen la finalidad de lograr una efectiva especificación de requisitos, aplicando técnicas y métodos de calidad.
- Propuestas relacionadas a mejorar el proceso de la ingeniería de requisitos ( (Horkoff & Yu, 2016), (Shah & Patel, 2014), (Saeed, et al., 2016), (Sankhwar, et al., 2014)). Estos trabajos pretenden mejorar la eficiencia en el proceso de la ingeniería de requisitos por medios de diversas técnicas y herramientas.
- Una de las más novedosas propuestas es la aplicación de técnicas de reúso al ámbito de los requisitos.
  - Para que las técnicas de reúso tengan relevancia, se dieron muchos estudios sobre el mejoramiento de las técnicas de indexación y recuperación de la información ( (Campos, et al., 2015), (Rini & Govilkar, 2016), (Kanchebua & Anand, 2016), (Lamarca, 2017), (Baeza-Yates & Ribeiro-Neto, 1999), (Garshol, 2004), (Zobel & Moffat, 2006)).
  - La aplicación de los patrones de Alexander (Alexander, et al., 1977), llevado al ámbito de la informática y en especial al ámbito de la Ingeniería de Requisitos tiene gran notoriedad. En este contexto tenemos muchos trabajos desarrollados. Propuestas de patrones como conjunto de modelos generales de requisitos (patrones de análisis) de los diferentes aspectos del negocio de una empresa (Fowler, 2017). Patrones como lineamientos o modelos de requisitos de software para su respectivo reúso ( (Vavassori & Cezario, 2013), (Withall, 2014), (Toval, et al., 2002), (Nicolás, et al., 2003), (Laguna, et al., 2001), (Durán, et al., 2000)).
  - Otros trabajos, con sus aportaciones, van en la misma línea de aplicación del reúso para lograr eficiencia y eficacia en el proceso de la Ingeniería de Requisitos ( (Pacheco, 2016), (Robertson & Robertson, 2012), (Sutcliffe, et al., 2006), (Marrero, et al., 2008), (Konrad & Cheng, 2002)).

En concordancia con la aplicación de técnicas de reúso al ámbito de los requisitos, esta tesis doctoral propone desarrollar un modelo de activos y un modelo de control de calidad orientado al reúso eficiente y efectivo de requisitos.

Recordemos que MORORE permite la representación de tres niveles para el reúso de los requisitos: representación individual de requisitos, representación de conjuntos de requisitos (patrones de requisitos) y representación de estructuras de tipos (clasificaciones

# INTRODUCCIÓN

---

de tipos de requisitos). Asimismo, consideramos la definición y aplicación de factores y métricas para el control de calidad de los productos reutilizables, y una guía de un proceso de creación y uso de activos reusables, utilizando MORORE.

El nivel de representación individual de requisitos permite la especificación correcta y completa de los requisitos para su reúso en un proyecto específico. Esta especificación incluye una colección de atributos que según se requiera podrán ser seleccionados en el momento de integrar un requisito a un proyecto. Este nivel también permite representar otros elementos que complementen la información del requisito para un proyecto específico: riesgos o casos de prueba asociados al requisito, y también archivos relacionados con el requisito. Otro punto importante de este nivel es la parametrización de requisitos. La parametrización permitirá particularizar los datos variables de un requisito para facilitar su adaptación a un proyecto específico. La parametrización también tiene el propósito de facilitar la aplicación del ámbito de reúso del requisito a un mayor número de supuestos.

Para los patrones de requisitos, planteamos la representación de una asociación problema-solución. Esta asociación se compone de atributos que describen el problema recurrente y la solución respectiva. Asimismo, se detalla el conjunto de requisitos interrelacionados que se ha demostrado resuelven el problema recurrente del ámbito de los requisitos. Este nivel también incluye otros elementos relacionados con el patrón como archivos, páginas WEB, etc.

El nivel de estructuras de tipo de requisitos considera la representación de un patrón de estructuras que hayan sido usadas con éxito en la organización de requisitos para un proyecto específico. Consta de un conjunto de atributos descriptivos de la estructura que permitan la ubicación y aprendizaje para su uso respectivo. Asimismo, incluye la estructura de tipos de requisitos propiamente dicha y la representación de elementos externos asociados con la misma.

Además, proponemos la aplicación de un control de calidad para el reúso efectivo de los productos reusables de cada uno de los niveles del modelo. Este control de calidad consiste en la definición de factores y métricas que se apliquen en el proceso de creación y uso de los elementos reusables. En consecuencia, se garantizará un mínimo de calidad en los elementos reusables para su aplicación a nuevos proyectos.

Complementariamente al modelo, consideramos la definición de un proceso de reúso para los activos software de todos los niveles del modelo. Este proceso incluye dos bloques generales:

- Indexación: es el proceso que abarca desde la creación o identificación, adaptación para el reúso, control de calidad y almacenamiento de los activos de requisitos de MORORE.
- Recuperación: es el proceso que comprende desde la búsqueda, selección, recuperación e integración en un nuevo proyecto de los activos de requisitos de MORORE.



# INTRODUCCIÓN

La finalidad que pretendemos con la propuesta es contribuir a la mejora de la fase de especificación de la ingeniería de requisitos. Por lo tanto, al mejorar la ingeniería de requisitos contribuimos a la mejora de todo el proceso de desarrollo de software.

## 1.7 Validez de la solución

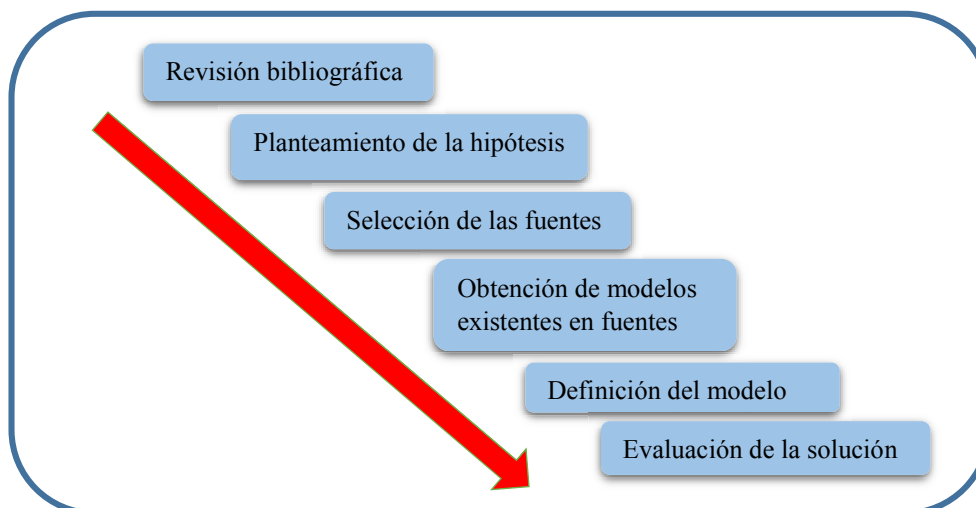
En todo proyecto de investigación existe la necesidad ineludible de su respectiva validación, es decir, la comprobación de que la solución propuesta resuelve el problema planteado. Para demostrar la validez de la propuesta que presentamos en esta tesis doctoral planteamos un experimento, descrito en el punto 4.1 Experimentación. Además, describimos otras actividades que conllevan al asentimiento de la comunidad científica relacionada con el presente trabajo de investigación. Podemos destacar entre las actividades: las publicaciones en revistas y congresos, conferencias en instituciones educativas, proyectos I+D, desarrollo de módulos informáticos, etc. Las otras actividades las detallamos en el punto 5.4 Aceptación.

El experimento, en forma general, consistió en comparar la aplicación del nivel de patrones de requisitos de MORORE (implementado en una herramienta automática), con formas tradicionales de desarrollo de requisitos (desarrollo de requisitos en MS-Word y en una herramienta CASE convencional). Para el efecto, trabajamos con un grupo 375 estudiantes de una universidad que desarrollaron requisitos para distintos proyectos. Un grupo de los estudiantes desarrolló requisitos bajo la propuesta de la Tesis y los otros dos grupos desarrollaron trabajos equivalentes usando MS-Word y una herramienta CASE tradicional. Luego se realizó un análisis estadístico de la información recogida y elaboramos las conclusiones de la presente tesis doctoral.

## 1.8 Metodología

En el desarrollo y validación de la investigación utilizamos métodos empíricos y conceptuales. Estos métodos son muy utilizados, y apropiados para investigación en ingeniería de software ( (Genero, et al., 2014), (Chavarriaga & Arboleda, 2004)).

En la Figura 1 hemos integrado el conjunto de pasos lógicos que permitieron definir y validar el desarrollo de la tesis propuesta.



*Figura 1 Proceso metodológico del desarrollo de la tesis.*

# INTRODUCCIÓN

---

## 1.8.1 Revisión bibliográfica

Como primer paso para la elaboración de la presente tesis doctoral hicimos una revisión de la bibliografía (Beltran, 2005) de los temas y trabajos relevantes relacionados a nuestra línea de investigación. De las fuentes revisadas encontramos como relevantes para el trabajo 138, distribuidas en 55 libros, 52 artículos, 2 reportes técnicos y 31 sitios WEB. Este estudio ayudó a especificar el problema de nuestro trabajo de investigación. Los principales puntos revisados son:

- Ingeniería de requisitos, abordamos definiciones, proceso y perfil de las personas involucradas en este proceso. De este estudio obtuvimos, principalmente, los activos específicos a reusar, debido a su importancia para las personas y el proceso de la ingeniería de requisitos.
- Reúso, analizamos las definiciones, beneficios y dificultades, y algunos ejemplos de activos reusables. El reúso es otro tema principal que motivó el desarrollo de la presente tesis doctoral.
- Indexación y recuperación de la información, analizamos las principales técnicas de indexación y recuperación de la información. Este estudio aportó en el desarrollo del proceso guía propuesto en esta tesis doctoral.
- Patrones, estudiamos las definiciones y características. Este estudio permitió evolucionar la idea principal de nuestro trabajo: conjuntos de elementos que han sido utilizados con éxito en la solución de problemas recurrentes en un ámbito determinado. En nuestro caso usamos la idea de patrón en el ámbito de la ingeniería de requisitos en el sentido específico de la presente investigación. También asimilamos las técnicas y procedimientos usados en la línea de patrones para abstraerlos al ámbito de los requisitos: plantillas de las características del patrón con sus respectivos atributos, características de aprendizaje del patrón, reúso de los patrones (recuperación e integración en un proyecto), elementos constitutivos del patrón, relaciones con otros elementos externos, etc.
- Calidad en la Ingeniería de Software, estudiamos los principales modelos de calidad propuestos en Ingeniería de Software. En base a este estudio planteamos el modelo de evaluación de la calidad de activos software del ámbito de la ingeniería de requisitos definidos en la presente tesis doctoral.
- Estándares de requisitos, revisamos los principales estándares relacionados con la ingeniería de requisitos. De esta revisión obtuvimos las diversas formas de especificación y las principales cualidades de calidad de los requisitos, que sirvieron en gran medida para la definición de MORORE.
- Herramientas CASE, probamos las principales herramientas informáticas para la gestión de requisitos. Los modelos de representación y formas de manejo de requisitos de estas herramientas, también fueron fuente importante de aporte para la definición de MORORE.
- Investigaciones relacionadas, revisamos otros estudios relacionados con la investigación de la presente tesis doctoral, haciendo una valoración crítica de las diversas investigaciones seleccionadas. Esta revisión permitió tener un mayor conocimiento del tema en estudio, que se sabe o que se desconoce al respecto, situar

en contexto nuestra investigación. Es decir, pudimos conocer el estado actual sobre el tema a investigar.

## **1.8.2 Planteamiento de la hipótesis**

En base al estudio de los temas relacionados en el actual trabajo de investigación formulamos la hipótesis. Esta hipótesis consiste en la creencia de que la solución planteada, reúso de activos software, en la presente tesis doctoral (MORORE) permitirá reducir el tiempo de desarrollo y mejorar la calidad de los requisitos, y de esta manera mejorar la especificación de la ingeniería de requisitos. La hipótesis fue explicada con mayor detalle en el acápite 1.4.

## **1.8.3 Selección de las fuentes**

Con el problema especificado, realizamos una búsqueda de la literatura haciendo uso de bases de datos bibliográficas: Scopus, ScienceDirect, ISI Web, Dialnet, etc. Escogimos los principales libros, estándares y artículos más relacionados con la línea de investigación del presente trabajo. Esta selección se hizo en base a los siguientes parámetros especificados (Guerrero, et al., 2012):

- Relación con el tema.
- Probabilidad que contenga modelos de activos, de control de calidad y de procesos.
- Obra destacada o autor destacado.
- Sugerencia de experto.
- Antigüedad no mayor a 30 años.

## **1.8.4 Obtención de los modelos existentes en las fuentes**

Con las fuentes de información seleccionadas, analizamos cada una, principalmente con el objetivo de obtener las diversas formas de representar y gestionar sus activos software. De este análisis obtuvimos:

- Las formas predominantes de especificar requisitos en forma individual y grupal. Asimismo, encontramos otras características que pudiéramos adaptar para el reúso de requisitos: plantillas, atributos de las plantillas, generación dinámica de las plantillas, parametrización, elementos asociados con el requisito, etc.
- Otro punto importante de los estudios realizados son las formas predominantes para organizar los requisitos. En este sentido analizamos también los respectivos entornos de aplicación de estas formas organizativas, otros elementos relacionados y los diferentes tipos de relaciones entre las clasificaciones de requisitos: jerarquía, asociación, dependencia, etc.
- También analizamos las diferentes formas que existen para controlar la calidad en el ámbito de la ingeniería de software. Este análisis condujo a la definición del modelo de factores y métricas para el control de calidad en el ámbito de la ingeniería de requisitos. Los valores propuestos fueron obtenidos en bases a un análisis de la literatura, experiencia del autor y opiniones de expertos en ingeniería de requisitos.

# INTRODUCCIÓN

---

- Asimismo, realizamos una revisión general sobre procedimientos y técnicas de reúso, específicamente sobre reúso de activos software. Y planteamos, como complemento al modelo, una guía de pasos generales para la creación y reúso de activos de software<sup>2</sup>.

## 1.8.5 Definición del modelo

Finalmente definimos MORORE en sus tres niveles de activos reusables: requisito individual, patrón de requisitos y patrón de estructura de tipos de requisitos. Empezamos por la definición de las formas de representación genérica de cada nivel de MORORE: especificación, características generales y particulares, etc. Luego definimos los factores y métricas de calidad para cada una de los niveles de MORORE en forma independiente. Por último, definimos el proceso general que permita la creación y reúso de requisitos aplicando MORORE.

El proceso de definición de MORORE se realizó en forma iterativa e incremental. De cada fuente revisada se obtenían características que se añadían a MORORE para incrementar su cobertura. Al final MORORE debe soportar todos los modelos revisados y otros posibles dentro del alcance específico descrito.

De manera específica se siguieron los pasos descritos en la Figura 2.

1. INICIO
2. Para cada fuente de información
  - 2.1. Revisar fuente
  - 2.2. Abstractar modelo existente en fuente
  - 2.3. Comparar con el modelo general MORORE
  - 2.4. Adicionar a MORORE nuevas características de modelo comparado
  - 2.5. Repetir hasta agotar fuentes
3. Adicionar a MORORE otras características ideadas
4. Optimizar MORORE
5. FIN

*Figura 2 Pasos para la definición de MORORE.*

## 1.8.6 Evaluación de la solución

Con el modelo definido, iniciamos un experimento que demuestre la validez de la hipótesis propuesta en la presente tesis doctoral. Este experimento está ya mencionado en el apartado 1.7 (Validez de la solución) y detallado en el capítulo 4 (Evaluación de la solución).

Para evaluar la validez de la hipótesis, se diseñó y ejecutó un experimento. Este experimento consistió en probar la aplicación de activos de MORORE (patrón de

---

<sup>2</sup> Un activo software, también llamado artefacto software, es un entregable del proceso de desarrollo de software. Por ejemplo: código ejecutable, componentes, especificación de requisitos, documentación, etc.

# INTRODUCCIÓN

---

requisitos), en el proceso de la ingeniería de requisitos con respecto a procedimientos convencionales. Este experimento fue posible utilizando la Herramienta swREUSER (swReuser, 2017), desarrollada por la compañía The Reuse Company. En esta herramienta se desarrolló un módulo que permite el reúso de patrones de requisitos basado en MORORE.

## 1.9 Limitaciones y amenazas de la investigación

A continuación, describimos algunos requisitos asociados al presente trabajo de investigación y que restringen el ámbito del estudio:

- El experimento se realizó con alumnos del pregrado de 3 titulaciones. Los resultados pueden tener alguna variación al realizarse el estudio con ingenieros expertos.
- El proceso propuesto como guía para el ingeniero abarca específicamente la parte de indexación de un activo de requisitos y la recuperación del mismo para ser reusado en un nuevo proyecto. No incluye todo el proceso de desarrollo de requisitos de un proyecto.
- Asimismo, el proceso no propone técnicas específicas nuevas para la indexación o recuperación de los activos. La propuesta del proceso se basa principalmente en una guía de pasos generales a seguir por el ingeniero para el reúso de un activo. Para la validación se usaron técnicas específicas comunes de indexación y recuperación.
- Los datos específicos del modelo para medir la calidad de MORORE, fueron obtenidos en base a un análisis de la literatura, experiencia del autor y opiniones de 4 expertos en ingeniería de requisitos. Dependiendo de otros factores particulares, estos valores podrían ser modificados configurando la futura herramienta de soporte de MORORE, pero el modelo seguiría siendo válido.

A pesar de las posibles consecuencias de las restricciones expuestas, el modelo y las aportaciones descritas en la presente tesis siguen siendo vigentes.

## 1.10 Estructura de la tesis doctoral

La presente memoria se ha estructurado en los capítulos que a continuación describimos:

- Capítulo 1. Introducción: capítulo actual de la tesis, en el que especificamos el objetivo, metodología de trabajo y contexto de la presente investigación. Exponemos el problema e importancia del mismo. Asimismo, hacemos una aproximación a la solución, para luego terminar con una breve descripción de la evaluación de la tesis doctoral y la estructura de la memoria.
- Capítulo 2. Estado de la cuestión: capítulo donde se describe el análisis de los temas de investigación directamente relacionados con la tesis doctoral y su aporte a la misma. También, describimos la revisión crítica de los principales trabajos existentes relacionados con el presente trabajo de investigación.
- Capítulo 3. Solución propuesta: hacemos una descripción exhaustiva del trabajo realizado: el desarrollo del modelo MORORE.

# INTRODUCCIÓN

---

- Capítulo 4. Evaluación de la solución: describimos el experimento realizado que permiten demostrar la validez de la hipótesis propuesta en la presente tesis doctoral.
- Capítulo 5. Conclusiones: Este capítulo expone las conclusiones extraídas en la realización de esta tesis doctoral. Además, se proponen futuras líneas de investigación surgidas a partir del trabajo de investigación realizado.
- Capítulo 6. Referencias: Se enumeran las referencias de las distintas fuentes de información consultadas y utilizadas para el desarrollo del presente trabajo de investigación.
- Anexos: En este apartado colocamos los elementos relevantes que permiten comprender mejor algunos puntos descritos en la presente tesis doctoral.

# ESTADO DE LA CUESTIÓN

## 2 Estado de la cuestión

En este apartado mostramos el estado del arte de las áreas o temas relacionados con la presente tesis doctoral. Asimismo, presentamos otros trabajos de investigación de la comunidad científica relacionada, que persiguen el mismo interés que ha motivado al presente trabajo: mejorar la ingeniería de requisitos.

Empezamos presentando el tema en el cual se enmarca la presente investigación: la ingeniería de requisitos. Tema donde encontramos la problemática y la motivación para el presente trabajo. Continuamos con los temas de Reúso y Patrones que sirvieron de base para las ideas para el planteamiento de la solución propuesta en esta tesis doctoral: reusar activos de la ingeniería de requisitos. Para que sea evidente el proceso de reúso, debe apoyar en técnicas de ingeniería de la información: Indexación y Recuperación. Proponemos como guía para los ingenieros, un conjunto de pasos generales que permiten almacenar y luego recuperar los activos software. Aunque en la tesis no proponemos técnicas específicas nuevas de indexación y recuperación, si sugerimos alguna común como trabajo futuro, por ello en el estado del arte hacemos una revisión de la Ingeniería de la Información. En MORORE proponemos qué para reusar efectivamente activos de requisitos, éstos últimos deben ser almacenados con un mínimo de calidad. Por tal motivo exponemos el tema de Calidad en la Ingeniería de Requisitos que permitió idear el modelo de calidad planteado en MORORE. Asimismo, revisamos distintos estándares relacionados con la ingeniería de requisitos. De estos estándares obtuvimos muchas características de especificación y organización, y cualidades de calidad de los requisitos. Estos elementos sirvieron de base para el desarrollo de los modelos de representación y calidad de MORORE que soportarán las características de dichos estándares y otras previsibles. En MORORE planteamos de modo complementario un proceso general que guíe a los ingenieros en el proceso de almacenamiento y reúso de los activos. La forma de especificación de este proceso se basó principalmente en la forma de especificación de procesos del PMBoK (PMBoK, 2013), que también exponemos en el estado del arte. También hicimos una revisión de un grupo de herramientas automáticas de gestión de requisitos, pues el desarrollo de las características de MORORE fue pensado para que sea soportado por un software. Por último, presentamos otros trabajos de investigación que también se orientan por las técnicas de reúso para el mejoramiento de la ingeniería de requisitos y las respectivas diferencias con el presente trabajo de investigación.

### 2.1 Ingeniería de requisitos

El motivo de la presente tesis doctoral es el desarrollo de un modelo para el reúso de requisitos. Este modelo es una propuesta para mejorar el proceso de la ingeniería de requisitos. En tal sentido la ingeniería de requisitos es la disciplina que está directamente relacionada con la presente investigación. En consecuencia, en el presente acápite mostramos algunos conceptos básicos respecto a la ingeniería de requisitos de donde obtuvimos las bases iniciales para el desarrollo de MORORE. Además, en base a este estudio proponemos una definición propia que contribuya al mejoramiento de esta disciplina.



## 2.1.1 Definiciones de ingeniería de requisitos

Se dice que en el presente vivimos en la era de la información y del conocimiento ( (Burch, 2005), (Trejo, 2016)). La economía ya no se basa en la producción de bienes o servicios tangibles, materias primas o mano de obra. En la actualidad la información y el conocimiento son los bienes intangibles que mueven el mundo socio-económico (Cobarsí-Morales, et al., 2013).

El control y aprovechamiento de estos bienes intangibles, a su actual magnitud, no hubiera sido posible sin el desarrollo de la informática. En consecuencia, la demanda de productos informáticos es abrumadora y supera claramente a la oferta.

Esta diferencia entre oferta y demanda mencionada compete directamente a la ingeniería de software. La ingeniería de software continúa en un proceso de maduración. Cada día surgen nuevos avances que mejoran gratamente esta disciplina, pero aún queda camino que recorrer para obtener resultados aceptables.

Con el objetivo de mejorar la ingeniería de software, muchos estudios se orientan actualmente a una de las fases de mayor impacto sobre el proceso de desarrollo de software: fase de análisis de requisitos ( (Saeed, et al., 2016), (Sankhwar, et al., 2014), (Génova, et al., 2017), (Govindaraju & Wiradanti, 2015), (Génova, et al., 2013), (Pressman & Maxim, 2015), (Sommerville, 2016), (Laplane, 2013), (Horkoff & Yu, 2016), (Sánchez, et al., 2015), (ISO/IEC/IEEE-29148, 2017)). Debido a la importancia de esta fase, se ha desarrollado toda una disciplina: la ingeniería de requisitos (IR).

La IR se encarga del estudio y mejoramiento de la fase de análisis, una de las fases más importantes del proceso de desarrollo de software (SDP: *Software Development Process*). Esta fase es un punto esencial del SDP a la cual muchas veces se resta importancia, y por tal, se pueden tener muchos inconvenientes. Es más, la mayoría de los fracasos cuando se desarrolla un software son debidos a una deficiente especificación y gestión de los requisitos ( (Shah & Patel, 2014), (Manoel, et al., 2013), (Ibañez & Rempp, 1996), (Standish, 2016), (Pressman & Maxim, 2015), (Sommerville, 2016), (Braude & Bernstein, 2016)).

Por tanto, en este apartado, y con el propósito de facilitar la comprensión de la tesis, mostramos algunas nociones relacionadas con la ingeniería de requisitos. Primero recogemos la definición de “requisito” más difundida y estandarizada publicada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE: Institute of Electrical and Electronics Engineers) (IEEE, 1990)<sup>3</sup>:

- (1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

---

<sup>3</sup> El estándar no especifica si estos tres enunciados son alternativos o complementarios de una sola idea. Para la presente investigación asumimos que estos enunciados se complementan para definir todo el ámbito de la definición de requisito.

## ESTADO DE LA CUESTIÓN

---

- (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Y
- (3) Una representación documentada de una condición o capacidad como en (1) o (2).

A continuación, podemos citar algunas de las principales definiciones de ingeniería de requisitos de diferentes autores y que sirven como referencia a muchas investigaciones:

Según Thayer (Thayer & Dorfman, 1997): “La ingeniería de requisitos facilita los mecanismos adecuados para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional.”

Según Loucopoulos y Karakostas (Loucopoulos & Karakostas, 1995): “La ingeniería de requisitos se encarga de todas la actividades que intentan comprender las necesidades exactas de los usuarios de un sistema software y traducir tales necesidades a especificaciones precisas y no ambiguas para que posteriormente puedan ser usadas en el desarrollo de un sistema.” Y “La ingeniería de requisitos puede ser definida como el desarrollo sistemático de los requisitos a través de un proceso iterativo y cooperativo en el que se analiza el problema, se documenta el resultado en diversos formatos de representación, y se comprueba la exactitud de la comprensión alcanzada.”

Según Zave (Zave, 1997): “La ingeniería de requisitos es la rama de la ingeniería de software que tiene que ver con las metas del mundo real para proveer servicios y restricciones en un grande y complejo sistema de software. También concierne a las relaciones entre los factores para la especificación precisa del comportamiento del sistema, y para su evolución de familias de sistemas.”

Podemos apreciar que en todas estas definiciones se hace referencia a la *comprensión de las necesidades*. Debemos de tener en cuenta que estas necesidades pueden ser originadas por un problema que hay que solucionar o por un objetivo o meta que el usuario desea alcanzar. Apreciamos también que en las definiciones citadas se menciona la importancia de la especificación de una solución propuesta. Y en forma particular Loucopoulos (Loucopoulos & Karakostas, 1995) habla del proceso evolutivo de los requisitos y por su parte Thayer (Thayer & Dorfman, 1997) menciona el aspecto de la gestión de los requisitos.

Teniendo en cuenta la definición de requisito de la IEEE (IEEE, 1990), las opiniones de los autores citados, y otros estudios y experiencias realizadas, planteamos una definición de la ingeniería de requisitos con su análisis respectivo. Con la definición propuesta pretendemos cubrir todo el ámbito de la ingeniería de requisitos según nuestra visión:

*“Es la rama de la ingeniería de software que se encarga de la captación, análisis, verificación y validación de la información relativa a las necesidades de los usuarios; de la invención de la especificación correcta y completa de los requisitos propiamente dichos, dentro de un proceso evolutivo y de negociación; y del mantenimiento o gestión*

## ESTADO DE LA CUESTIÓN

---

*de los requisitos dentro de todo el proceso de desarrollo de software. El producto o activo del proceso de la ingeniería de requisitos es el documento de especificación de requisitos”.*

Conocemos que el campo de acción de la ingeniería de software es todo el SDP, el cual es un proceso iterativo e incremental, hasta lograr un software de calidad. En las etapas iniciales de este proceso se empieza la fase de análisis de requisitos, que por su complejidad e importancia para el éxito de todo el proyecto se ha formalizado en lo que actualmente denominamos ingeniería de requisitos. En consecuencia, al igual que Zave (Zave, 1997), mencionamos que la ingeniería de requisitos es una rama de la ingeniería de software.

Todo este proceso de la ingeniería de requisitos naturalmente debe empezar por la recopilación de la información, la cual después de ser analizada debe pasar por la verificación respectiva de los ingenieros de requisitos y la validación de parte de los usuarios. Todo este subproceso constante de captación de información sirve de base para las siguientes fases de la ingeniería de requisitos.

En la definición propuesta mencionamos también que la ingeniería de requisitos se encarga de *la comprensión de las necesidades de los usuarios*. Sostenemos, al igual que el IEEE (IEEE, 1990), que estas necesidades se refieren a la condición problemática o capacidad requerida para alcanzar una meta, por ejemplo: la demora en la facturación de una tienda, que el cliente de un banco obtenga fácilmente información sobre sus cuentas por internet, poder controlar la información laboral y profesional del personal de una organización, etc.

Aunque consideramos que estas necesidades o deseos de los usuarios van más allá del ámbito informático, pensamos que la ingeniería de requisitos debe abarcar la comprensión de las mismas. Y de esta manera, luego poder crear, especificar y gestionar los requisitos propiamente dichos de manera adecuada. En este sentido cabe destacar que las necesidades se descubren, la descripción estructurada de requisitos se inventa. Los requisitos no están ahí esperando que alguien los descubra, sino que creamos, construimos o inventamos su especificación en un proceso interactivo entre el cliente y el ingeniero (Génova, et al., 2017). Estos requisitos propiamente dichos son las capacidades o funciones que deberá tener el futuro software para satisfacer las necesidades del usuario y que deberán ser especificados de forma correcta y completa (verificados y validados).

También especificamos el carácter evolutivo y de negociación de los requisitos. Se sabe que se firman contratos para el desarrollo de un software, y que se pretende especificar los requisitos de forma detallada. Pero en la práctica es imposible pensar en un contrato inicial con todos los detalles que tendrá el software. Existen siempre muchos problemas en los proyectos de software por la volatilidad de los requisitos ( (Shah & Patel, 2014), (Manoel, et al., 2013), (Standish, 2016), (Braude & Bernstein, 2016), (Ibañez & Rempp, 1996), (Christel & Kang, 1992)); los clientes desean que se agreguen o modifiquen funcionalidades al mismo precio y el ingeniero insiste en que esas modificaciones no estaban presupuestadas en el contrato inicial.

Como ya hemos mencionado, es natural que los requisitos cambien durante el desarrollo del software. En este sentido para llevar con éxito un proyecto, aparte de

sobrellevar la complejidad del desarrollo del mismo, es preciso la complicitad y el contacto continuo entre el cliente y el equipo de desarrollo. De esta forma ambos conocerán las dificultades de ambas partes y trabajarán unidos para solucionarlas. En consecuencia, si la sinergia cliente-ingeniero funciona bien, los requisitos no tendrán que ser motivo de discusión contractual sino parte natural de la metodología o proceso de desarrollo.

Otro punto importante es la adecuada gestión de los requisitos. Esta gestión es probablemente el factor más relevante para el éxito o fracaso de un proyecto. En la práctica conocemos que los requisitos son cambiantes, es imposible conocer al inicio de un proyecto todos los detalles del futuro software, es natural que los requisitos se transformen dentro del proceso iterativo y evolutivo del software. En consecuencia, es de suma importancia que se tenga una buena especificación y organización de los requisitos, así como una adecuada gestión de los mismos.

Por último, en la definición propuesta de ingeniería de requisitos hablamos de la documentación. La documentación es el resultado del proceso de la ingeniería de requisitos y por consiguiente se convierte en uno de los activos del proceso de desarrollo de software. En la actualidad se usan herramientas de gestión de requisitos, las cuales deben tener las respectivas funciones para generar los documentos de las versiones evolutivas de requisitos que permitan la gestión de los mismos.

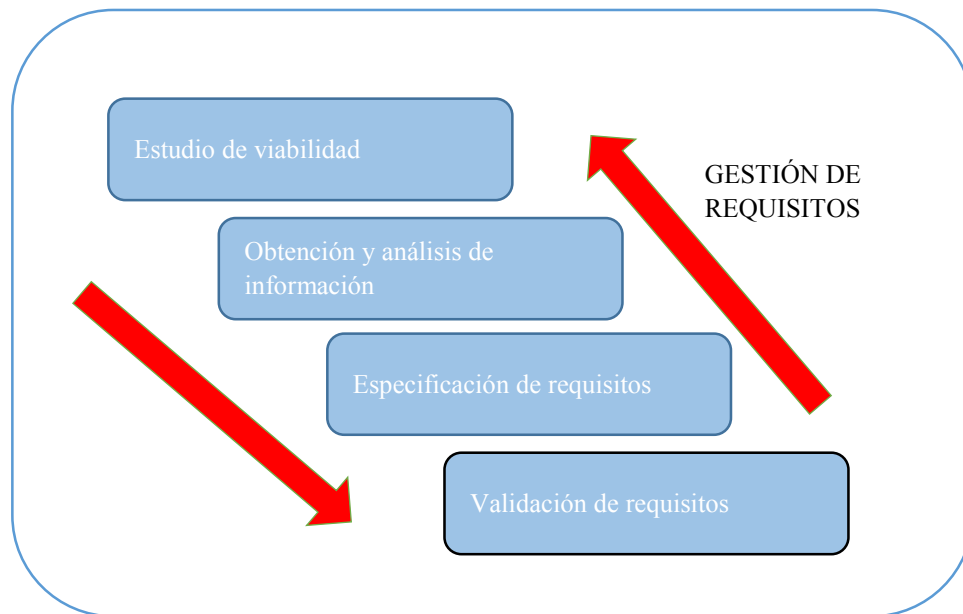
### **2.1.2 Proceso de la ingeniería de requisitos**

El proceso de la ingeniería de requisitos (REP) comienza al inicio de un proyecto de software y se extiende durante todo el SDP (SDP: software development process) (Horkoff & Yu, 2016), (Sankhwar, et al., 2014), (Sommerville, 2016), (Pressman & Maxim, 2015), (Braude & Bernstein, 2016),). El resultado del REP es el documento de especificación de requisitos en sus respectivas versiones, el cual es un activo del SDP. En base a la definición propuesta, al REP lo podemos dividir en fases estructuradas bien definidas, las cuales se desarrollan de manera iterativa e incremental, como apreciamos en la Figura 3.

- Estudio de viabilidad. Estudio rápido que permite definir a alto nivel la funcionalidad y ver si es factible su realización dentro de los límites económicos y de tiempo establecidos. Resultado: informe de viabilidad.
- Obtención y análisis de información. Proceso orientado a obtener las necesidades y los requisitos mediante observación de sistemas existentes, entrevistas con usuarios y proveedores, análisis de tareas, etc. Resultados: necesidades y modelos del dominio y del sistema (Génova, et al., 2009).
- Especificación de requisitos. Proceso de traducción de la información recopilada y creada en un documento que define un conjunto de requisitos. Resultados: requisitos de usuario y del sistema (software).
- Validación de los requisitos. Esta actividad comprueba la veracidad, consistencia y completitud de los requisitos. Resultado: documento de requisitos.

# ESTADO DE LA CUESTIÓN

- Gestión de los requisitos. Esta actividad se desarrolla a lo largo de todo el SDP, permite la actualización de los cambios constantes que sufren los requisitos. Resultado: versiones del documento de requisitos.



*Figura 3 Proceso de la Ingeniería de Requisitos.*

Existen muchos estándares, métodos, herramientas y técnicas (Entre otros, mencionamos: (INCOSE, 2017), (ISO/IEC/IEEE-29148, 2017), (Abushark, et al., 2016), (Ito, 2016), (CORE, 2017), (DOORS, 2017)) que dan soporte a las diferentes fases de la ingeniería de requisitos. Pero consideramos que hace falta mejorar mucho estos elementos tecnológicos y también aumentar la concienciación de la importancia de la ingeniería de requisitos por parte de los involucrados en el SDP. En este sentido, como ya mencionamos, MORORE plantea, mediante el reúso, facilitar a los ingenieros la especificación de requisitos en cuanto a calidad y eficiencia.

## 2.2 Reúso

Para lograr el propósito de la presente tesis doctoral proponemos el uso de técnicas de reúso. Estas técnicas las aplicamos al ámbito de la ingeniería de requisitos donde adquieren mayor valor que en fases más avanzadas del proceso de desarrollo de software. En este sentido, también el tema del reúso está alineado con el desarrollo de la presente investigación descrita en la tesis doctoral. A continuación, presentamos algunos conceptos acerca del reúso que permitieron el desarrollo de la presente tesis doctoral.

### 2.2.1 Definición de reúso de software

Hoy en día las organizaciones buscan mejorar la calidad y hacer eficientes los procesos de fabricación de sus productos. En este sentido se puede apreciar una práctica común: el usar productos ya existentes. A este proceso se denomina reúso, el aprovechamiento de los productos disponibles para mejorar la calidad, aumentar la productividad y disminuir el coste de fabricación de los nuevos productos ( (TRC, 2017), (Llorens, 1996)). Por su parte, Fraga (Fraga, 2010) define un marco de trabajo y una metodología para la reutilización en general de cualquier conocimiento a bajo coste.

En el ámbito de la informática y en especial en el dominio del software, la aplicación del reuso va tomando mucha relevancia. Krueger define reuso de software como el proceso de creación de sistemas de software a partir de un software existente, en lugar de tener que diseñarlo desde el principio (Krueger, 1992). Por su parte Prieto-Díaz y Freeman (Prieto-Díaz, 1987) definen el reuso de software como cualquier procedimiento que produce o ayuda a producir un sistema mediante el nuevo uso de algún elemento procedente de un esfuerzo de desarrollo anterior. Pero es precisamente en la ingeniería de software, donde la reutilización adquiere mayores dificultades. Estas dificultades van desde la imposibilidad de realizar desarrollos informáticos mediante la unión directa de componentes software, por lo relativamente joven que es la disciplina de la ingeniería de software (cambios constantes que dificultan el reuso), la gran producción de activos o artefactos que se pueden reusar (dificultades en los métodos de recuperación), etc. Pero a pesar de esto, debido a los avances tecnológicos en el ámbito del reuso, esta técnica es una de las más prometedoras para el mejoramiento de la ingeniería de software (Parra, 2016), (Sommerville, 2016), (Llorens, 1996)).

En este sentido van desapareciendo los *programadores artesanos*<sup>4</sup>, que pretenden crear todo desde cero cada vez que inicia un proyecto de software. El desarrollo de sistemas de informáticos está dejando de ser un arte para convertirse en una ingeniería. Existen ya muchos estándares a nivel mundial que unifican lenguajes, productos y procedimientos de la ingeniería de software (UML, 2017), (ISO/IEC-14598-1, 2017), (ISO/IEC-12207, 2017)). Herramientas y técnicas que facilitan la gestión y también el reuso de activos software (Fayad & Flood, 2016), (Tahir, et al., 2016), (Moros, et al., 2008)). Además, en la actualidad las técnicas de recuperación facilitan el reuso (Büttcher, et al., 2010). Por ejemplo, investigaciones combinando el procesamiento del lenguaje natural y las tecnologías WEB (Rini & Govilkar, 2016) traen muchos avances en la recuperación de la información. Asimismo, la explotación de la información temporal de documentos (Campos, et al., 2015), (Kanhbua & Anand, 2016)) ha permitido mejorar la eficacia de la recuperación de la información.

### 2.2.2 Beneficios del reuso de software

Es innegable los grandes beneficios que produce el reuso del software, son muchos los estudios que lo avalan (Rajani & Rama, 2016), (McIlroy, 1968), (Boehm, 1993), (Boehm, 1999), (Kotonya, 1998), (TRC, 2017)). Estamos de acuerdo con Sommerville, que los beneficios del reuso aumentan considerablemente en la medida que los activos son más distantes del código (Sommerville, 2016). Si los activos software reusables son de un mayor nivel de abstracción (por ejemplo: de diseño o de análisis), están más independizados de la plataforma y por lo tanto de los riesgos que ésta conlleva, por ejemplo, la obsolescencia antes del reuso del activo. Estos beneficios están enmarcados en las siguientes líneas (TRC, 2017):

- Reducción en los costes de desarrollo.

---

<sup>4</sup> Bassett denomina programadores artesanos, a aquellos profesionales software que no tienen en cuenta los conceptos de la reutilización software, y los asemeja a fabricantes de coches que para construir un modelo necesitan reinventar la rueda (Bassett, 1987).

- Aumento de la calidad de los productos.
- Aumentos de la productividad de los ingenieros de software.
- Mejoras en las actividades de mantenimiento.
- Mejoras en las actividades de control y planificación.

### 2.2.3 Dificultades del reúso de software

A pesar de los grandes beneficios que produce el reúso y en especial para el reúso de software, estas prácticas aún no están muy desarrolladas. Existen muchas investigaciones que hacen posible a cierto nivel el reúso de software, pero tenemos aún muchas dificultades que superar. A continuación mencionamos algunas de las dificultades que afronta el reúso de software ( (Pressman & Maxim, 2015), (TRC, 2017)):

- Aún son pocas las organizaciones que tengan una planificación seria, prioritaria y exhaustiva para el reúso de software.
- Existe un número creciente de herramientas y componentes que proporcionan un soporte directo para el reúso de software. Pero la mayoría de los ingenieros y organizaciones implicados en el proceso de desarrollo de software no utilizan estas herramientas para el soporte al reúso de software.
- Aun no podemos decir que existe una cantidad aceptable de entrenamiento disponible para ayudar a los ingenieros de software y a los administradores a comprender el reúso de software.
- En los programas de formación en ingeniería de software son escasos los vínculos que incluyen el reúso de software de manera exhaustiva.
- Existen muchos especialistas detractores del reúso de software. Estos especialistas continúan con la creencia que el reúso es más un problema que un beneficio.
- Existen muchos ingenieros que propugnan las metodologías de desarrollo de software que dificultan el reúso.
- Es escasa la inversión en el reúso del software. Las organizaciones no apuestan por la generación y almacenamiento de activos software.

Como apreciamos, estas dificultades se producen principalmente por falta de confianza en el reúso de software. Se sabe que los beneficios del reúso de software son grandes, pero aún se piensa que son mayores los problemas para lograr un reúso de software efectivo.

### 2.2.4 Activos Software reusables

Recordemos que un activo software es una pieza tangible de información resultante del proceso de desarrollo de software. En la presente tesis doctoral planteamos que cualquier activo software es factible de ser usado más de una vez en diferentes proyectos de software similares. También tengamos presente que el código fuente no es la única forma de reusar software, es más, no es el único activo software que se pueda reusar de forma eficiente en el proceso de desarrollo de software ( (Sommerville, 2016), (Pressman & Maxim, 2015), (TRC, 2017)).

## ESTADO DE LA CUESTIÓN

---

A continuación planteamos algunos ejemplos de activos software que pueden ser reusados en diferentes proyectos (Mili, et al., 2002):

- Código ejecutable. Por ejemplo, las funciones representadas en forma de componentes legibles e indexados por el significado de sus propiedades funcionales.
- Código fuente. Por ejemplo, los componentes que incorporan funciones e información estructural, representados por lenguajes de programación e indexados por el significado de sus propiedades estructurales y por sus propiedades funcionales.
- Especificación de requisitos. Por ejemplo, especificación de las necesidades de los usuarios, representada en lenguaje natural, en notación formal (lógica, sistemas axiomáticos y lenguajes formales) o ambos. Indexado por el significado de propiedades funcionales que los requisitos abarcan y puede ser reusado para construir cualquier especificación compuesta o variaciones en el producto original.
- Diseños. Los diseños son representaciones genéricas de decisiones que encierran el conocimiento de resolución de problemas. Estos diseños capturan la información estructural más que la información funcional. Los diseños están representados por patrones e indexados por las características de los problemas que ellos solucionan.
- Documentación. Documentación en lenguaje natural que puede acompañar a los otros activos reusables (especificación, diseño, etc.). Representados en documentación en lenguaje natural e indexados por atributos del documento, técnicas de indexación y herramientas relacionadas.
- Arquitecturas. Módulos de arquitectura de software (conjunto de componentes que intercambian datos). Representados por el significado de notación especializada e indexados por el significado de sus características arquitecturales.

### 2.3 Indexación y Recuperación de la Información

Según Manning (Manning, et al., 2008), la recuperación de la información (IR: Information Retrieval) se ocupa de encontrar material (normalmente documentos) de una naturaleza no estructurada (normalmente texto) que se encuentra en grandes colecciones (normalmente almacenada de forma digital) y satisface una necesidad de información. Podemos denotar en esta definición la acción de “búsqueda” y la información a buscar.

En cuanto a la búsqueda, entramos en el ámbito de las consultas que son comparadas con un índice para encontrar el material buscado (Büttcher, et al., 2010), (Croft & Lafferty, 2003)). Como vimos en la definición de IR, el material a buscar es no estructurado, por lo cual, desechemos las búsquedas a bases de datos relacionales u similares, que contengan información estructurada. El material buscado, escrito en lenguaje natural (no estructurado), típicamente es ubicado por similitud a una necesidad de información plasmada en una consulta.

Para aclarar el proceso de recuperación de información Seco (Seco, 2009) plantea un esquema plasmado en la Figura 4. En este esquema podemos apreciar los elementos más importantes que intervienen en el proceso de la recuperación de la información: La comparación entre la representación de la información a buscar (objetos indexados: indexación) y la representación de la necesidad de información (Consultas: recuperación).



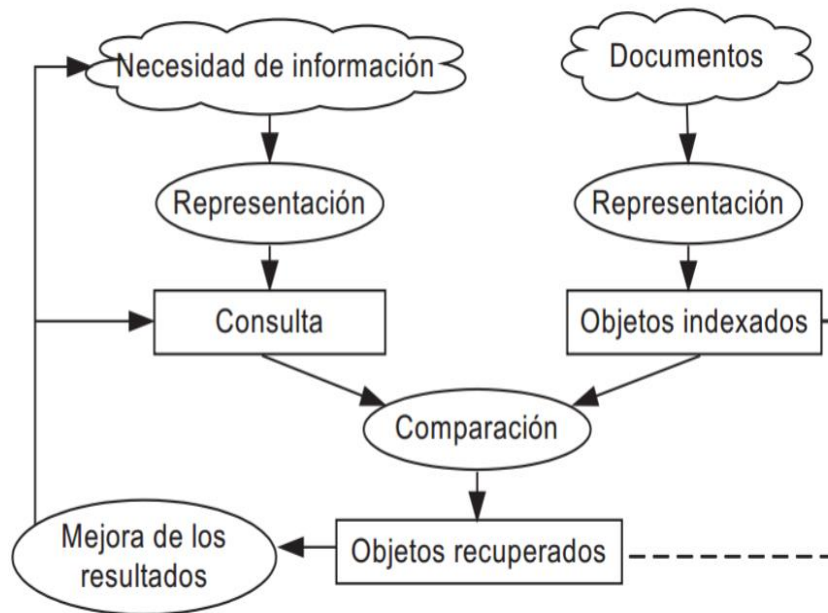


Figura 4 Esquema de proceso de recuperación de información. Fuente: (Seco, 2009)

Asimismo, apreciamos en la Figura 4, tareas representativas de la IR: la forma de representación de la información y la representación de la consulta, la comparación y la forma de medir la recuperación propiamente dicha.

### 2.3.1 Modelos de representación de la información

En el proceso de recuperación de información, todo comienza con la necesidad de información que se tenga. Esta necesidad debe ser planteada a modo de consulta (conjunto de palabras específicas que representen la necesidad). Esta consulta debe ser comparada con la representación de la información que se tenga almacenada (índice). Luego ser seleccionada y recuperada.

Pero para para que se pueda realizar la comparación de una consulta con un índice, el índice debe ser creado. Este índice es el resultado de una transformación lógica de la información almacenada. A esta transformación se denomina indexación.

La indexación está muy relacionada a las consultas, por ello, antes de hablar de indexación es necesario ver diferentes tipos de consultas. Podemos distinguir tres tipos principales de consultas: las consultas basadas en términos clave (comparación con términos claves de forma individual empleando operadores lógicos), las consultas de reconocimiento de patrones en texto (usan algoritmos más complejos y son más utilizados como complementos de las consultas con términos claves) y las consultas sobre la estructura de los textos (son más específicas y dependen de la estructura del documento que contiene la información).

Centramos nuestra atención sobre los modelos de indexación relacionados a las consultas basadas en términos claves, que son las más comunes ( (Garshol, 2004), (Zobel & Moffat, 2006), (Lamarca, 2017), (Manber & Myers, 1990), (Baeza-Yates & Ribeiro-Neto, 1999)). Entre estas tenemos:

- Lista de términos. Estructura orientada a la palabra compuesta de: el vocabulario y la lista de ocurrencias (incluyen posición de la palabra). Entre estas tenemos: glosarios, diccionarios, índices.
- Clasificaciones y categorías. Se orientan a la categorización de la información en clases y subclases. Entre estas tenemos encabezamiento de materias, taxonomías.
- Listas relacionadas. Se orientan a las relaciones entre los términos. Consisten en términos o listas de términos y distintas relaciones entre los mismos. Entre estas tenemos: tesauros (términos y relaciones), ontologías (términos, relaciones, reglas y axiomas), redes semánticas (organización no jerárquica de términos, relaciones de diferentes tipos a modo de grafos).
- Mapas de conocimiento. Se orienta a la representación gráfica de la información. Consiste en una representación visual, mediante un diagrama o mapa. Tenemos, por ejemplo: los mapas conceptuales, representación visual de redes semánticas y grafos conceptuales.

### 2.3.2 Modelos de recuperación de la información

Teniendo la información estructura de alguna manera vista el punto 2.3.1 (Modelos de representación de la información) hay que realizar un búsqueda específica según alguna necesidad. Existen modelos de estrategias de recuperación de información que presentamos a continuación:

- Modelo Booleano. Es el modelo de recuperación más simple, basado en la teoría de conjuntos y el álgebra booleana. Por simplicidad y su formalismo fue usado por muchos de los primeros sistemas bibliográficos comerciales. Su estrategia está basada en un criterio de decisión binario (pertinente o no pertinente).
- Modelo Vectorial. El modelo de recuperación vectorial permite el emparejamiento parcial, asignando pesos no binarios a los términos índice de las preguntas y de los documentos. Estos pesos de los términos se usan para computar el grado de similitud entre cada documento guardado en el sistema y la pregunta del usuario.
- Modelo Probabilístico. Se basa en la verificación probabilística, dados un documento y una pregunta. Realiza un cálculo de la probabilidad de que ese documento sea relevante para esa pregunta.

Actualmente tenemos estudios que incluyen técnicas interdisciplinarias que enriquecen este tema. Por ejemplo, investigaciones combinando el procesamiento del lenguaje natural y las tecnologías WEB (Rini & Govilkar, 2016) traen muchos avances en la recuperación de la información. Asimismo, la explotación de la información temporal de documentos (Campos, et al., 2015), (Kanebua & Anand, 2016)) ha permitido mejorar la eficacia de la recuperación de la información.

MORORE define un proceso de indexación y de recuperación como complemento al modelo. Este proceso se sugiere como guía general de pasos al ingeniero para el reúso de los activos de propuestos en la tesis. No incluye las técnicas específicas vistas en este apartado, aunque se sugiere como trabajo futuro la creación de una herramienta automática con el uso de tesauros (listas relacionadas). Se debe especificar que la

implementación realizada para el experimento de la tesis, se hizo con listas de términos para la representación de la información y un modelo de recuperación booleano.

## 2.4 Patrones

En un proceso de desarrollo de sistemas de información, los ingenieros de software se enfrentan cada día a multitud de problemas de distinta magnitud. Los ingenieros con experiencia resuelven, de forma mayormente intuitiva, muchos de los problemas de modelado de sistemas reales. El mejor profesional es el que reutiliza la misma solución retocada para resolver problemas similares en situaciones distintas. Estas experiencias, que engloban técnicas y criterios experimentales efectivos y probados, pueden ser difícilmente transmitidas formalmente a ingenieros noveles.

En la actualidad la idea de reuso cobra gran relevancia, especialmente en la orientación a objetos. Esta idea se vuelve insuficiente, si sólo la aplicamos a nivel de componentes de software en el proceso global de desarrollo (TRC, 2017). Se necesita una forma de transmitir el conocimiento de estas experiencias efectivas de solución a niveles de abstracción más altos del proceso de desarrollo de software.

Christopher Alexander (Alexander, et al., 1977), desde el ámbito de la arquitectura civil, propone la idea de los *patrones*, la cual luego es trasladada a la ingeniería de software (Cunningham & Kent, 1988), (Coplien, 1992), (Fayad, et al., 2014)), destacando principalmente con los famosos patrones de diseño de Gamma (Gamma, et al., 1995). Los patrones pretenden ser la solución al problema que se plantea de la comunicación de experiencias. Cada patrón describe un problema recurrente en un entorno, para describir después el camino a la solución a ese problema, de tal forma que pueda ser reutilizado en proyectos distintos.

Esta idea de patrones, por su misma naturaleza genérica de Problema-Solución-Reutilización, se ha popularizado a nivel mundial (Buschmann, et al., 2007), (Larman, 2004)). En la actualidad existen muchas investigaciones al respecto, las cuales se orientan a muy distintos ámbitos de la actividad humana. Asimismo, dentro de la ingeniería de software se han extendido tanto que se han creado y siguen creando catálogos de patrones para problemas en distintos niveles de abstracción. Por ejemplo, estamos hablando ya no sólo de patrones de diseño, sino también tenemos patrones de programación, de arquitectura, de análisis, etc. En el presente apartado planteamos el estudio de los patrones que ha servido de base para la definición de los niveles de MORORE.

### 2.4.1 Definiciones de patrón

En la práctica del desarrollo de software los ingenieros, con grado de experiencia medio o alto, aplican normalmente técnicas intuitivas para dar solución a problemas determinados. Estas técnicas efectivamente aplicadas a problemas específicos son fruto de situaciones problemáticas similares que estos profesionales atravesaron anteriormente saliendo triunfantes. Los patrones proponen una forma de transmitir estas experiencias de soluciones exitosas a problemas recurrentes. A continuación, presentamos algunas definiciones que describen a los patrones, surgidas en el ámbito de la arquitectura civil y posteriormente trasladados a la ingeniería de software.

## ESTADO DE LA CUESTIÓN

---

Según Christopher Alexander (Alexander, et al., 1977): “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma.”

Esta definición propuesta por Alexander intenta identificar y resolver, en un marco descriptivo formal, problemas esenciales que se presentan en el dominio de la arquitectura. Los ingenieros de software han utilizado esta definición como catalizador de tendencias de desarrollo en el diseño orientado a objetos de sistemas informáticos. La definición expuesta por Alexander se refiere a patrones de ciudades y edificios, pero son ciertamente válidas en el ámbito del diseño orientado a objetos, por ejemplo, nuestras soluciones se expresan en términos de objetos e interfaces en vez de paredes y puertas, pero en esencia ambas se orientan a la solución de un problema dentro de un contexto específico.

Las definiciones descritas a continuación están basadas en la definición de Alexander y están referidas ya al ámbito del software.

Según Gamma (Gamma, et al., 1995): Un patrón de diseño nombra, abstrae e identifica los aspectos claves de una estructura de diseño común, que los hace útiles para crear un diseño orientado a objetos reusable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema concreto, describiendo cuándo aplicarlo y si tiene sentido hacerlo teniendo otras restricciones de diseño, así como las consecuencias, las ventajas e inconvenientes de su uso.

Según Buschmann (Buschmann, et al., 2007): Un patrón describe un problema de diseño recurrente, que surge en contextos específicos de diseño, y presenta un esquema genérico probado para la solución de este. El esquema de la solución describe un conjunto de componentes, responsabilidades y relaciones entre estos, y formas en que dichos componentes colaboran entre sí.

Estas últimas definiciones están referidas principalmente al dominio del software. De estas definiciones podemos resaltar ideas esenciales para la comprensión de un patrón y que a continuación especificamos:

- Un patrón de diseño, es un conjunto de información que describe una solución a un problema de diseño recurrente no trivial.
- La efectividad de esta solución ha debido ser probada muchas veces por profesionales expertos dentro de un contexto determinado en situaciones anteriores.
- Esta solución debe ser reusable, para problemas de diseño similares en distintas circunstancias.
- Esta descripción debe ser lo suficientemente explícita para que pueda ser transmitida a ingenieros no expertos.

Después de analizar las definiciones expuestas, abstraemos algunos conceptos importantes de los patrones en general:

# ESTADO DE LA CUESTIÓN

---

- Contexto. Conformado por el entorno, situación, o condiciones interrelacionadas dentro de las cuales existe el patrón.
- Problema. Esta dado por un asunto insatisfecho, algo que se necesita estudiar y resolver.
- Solución. Esta referida a la respuesta al problema dentro del contexto y muestra el camino para resolver las dificultades.
- Reutilizable. La descripción debe ser clara, expresada de manera formal, aunque se encuentre en un ámbito general, para que pueda ser transmitida y reutilizada por ingenieros no expertos.

## 2.4.2 Características de un patrón

El concepto de patrón, aunque tiene un carácter general, pensamos que debe contar con las siguientes propiedades para que sea eficientemente reutilizado.

- Debe ser una solución que forma parte del conocimiento de un experto.
- Debe facilitar la comunicación entre los ingenieros, promoviendo un vocabulario accesible.
- Debe facilitar el aprendizaje de ingenieros no expertos.
- La solución que describe ha debido ser probada más de una vez en distintos casos.
- Debe proveer una plantilla conceptual que muestra el núcleo de una solución a un problema específico.
- Debe describir elementos y relaciones entre ellos: módulos, estructuras y mecanismos complejos.

## 2.4.3 Descripción de un patrón

Para describir un patrón y que pueda ser reutilizado con éxito necesitamos representar las decisiones, alternativas, ventajas e inconvenientes que son su razón de ser. Normalmente, un patrón está documentado en forma de una plantilla, es una práctica común, pero no significa que sea la única forma de hacerlo. A continuación presentamos como referencia los elementos de descripción de patrones propuesto por Gamma (Gamma, et al., 1995) para el ámbito del diseño, pero generalizados para cualquier tipo de patrón.

- Nombre del patrón. Debe ser un nombre corto y significativo, generalmente una o dos palabras. El nombre pasará a formar parte de nuestro vocabulario técnico.
- Clasificación. Para los tipos de patrón no hay una clasificación formal, pero siempre se suelen organizar de alguna manera.
- Propósito u objetivo. Representado por una frase breve que describe el problema concreto y lo que hace el patrón para resolverlo.
- También conocido como. Especificación de otros nombres con los cuales se conoce al patrón.

## ESTADO DE LA CUESTIÓN

---

- Aplicabilidad. Describe en qué situaciones o escenarios se puede aplicar el patrón, ejemplos y formas de reconocer tales situaciones.
- Estructura. Se representa el patrón mostrando sus elementos y relaciones constitutivas. Cómo colaboran los elementos del patrón para cumplir el objetivo.
- Participantes. Especificación de los elementos del patrón.
- Consecuencias. Se especifican las ventajas e inconvenientes que conlleva el uso del patrón.
- Implementación. Descripción de las dificultades, trucos o técnicas que deberíamos tener presentes al momento de aplicar el patrón.
- Usos conocidos. Descripción de ejemplos donde haya sido utilizado el patrón.
- Patrones relacionados. Especificaciones de otros patrones con los cuales esté relacionado, las principales diferencias y los patrones con los que se debería usar.

### 2.4.4 Orientaciones de aplicación de un patrón

Por la naturaleza de la idea de patrones, éstos solucionan problemas que existen en muchos niveles de abstracción. En el ámbito informático, los patrones inicialmente fueron aplicados en la fase de diseño de los sistemas de información. Sin embargo, existen otros ámbitos de la ingeniería de software donde se puede aplicar el concepto genérico de patrón. Por ejemplo:

- Patrones organizativos. Describen la estructura y prácticas de las organizaciones humanas, especialmente las productoras de software.
- Patrones de análisis. Describen un conjunto de prácticas destinadas a elaborar modelos de los conceptos principales de la aplicación que se va a construir (Schweiger, 2013), (Withall, 2010)). La intención principal de estos patrones es ayudar a las personas que realizan el trabajo de modelado, pues no siempre tienen experiencia al respecto y, en la mayoría de los casos, construyen sus modelos sin referencia alguna.
- Patrones de arquitectura. Expresan un paradigma fundamental para estructurar u organizar un sistema software. Proporcionan un conjunto de subsistemas o módulos predefinidos, con reglas y guías para organizar las relaciones entre ellos (Buschmann, et al., 2007), (Kircher & Jain, 2004), ). Ejemplo: Capas o Layers (aplicaciones: JVM, API, Windows NT), Pipes and Filtres (aplicaciones: UNIX), Pizarrón o Blackboard (aplicaciones: Hearsay, Inteligencia Artificial).
- Patrones de diseño. Proporcionan un esquema para refinar los subsistemas o componentes de un sistema software y las relaciones entre ellos. Describen estructuras recurrentes para comunicar componentes que resuelven un problema de diseño en un contexto particular. Son patrones de un nivel de abstracción menor que los patrones de arquitectura. Están por lo tanto más próximos al código fuente final (Gamma, et al., 1995), (Larman, 2004)).
- Patrones de programación (*Idioms patterns*). Son patrones de bajo nivel, específicos de un lenguaje de programación determinado. Describen cómo implementar aspectos

particulares de los componentes de un patrón de diseño usando las características y potencialidades de un lenguaje de programación concreto (Nystrom, 2014).

La idea de los patrones ha adquirido gran popularidad. En el ámbito de la ingeniería de software se están desarrollando muchos proyectos de investigación al respecto. Como hemos visto, no sólo se está aplicando en los llamados patrones de diseño, sino que la idea está transportándose a otras actividades del desarrollo de software.

El concepto de patrón tiene carácter general y por consiguiente es aplicable a un sinnúmero de situaciones de diferente tipo, estilo, entorno, etc. En lo referente al software, los patrones están siendo utilizados con éxito en muchos ámbitos del proceso de ingeniería de software. En definitiva, están mejorando las prácticas en el desarrollo de sistemas de información.

Donde exista la posibilidad de aplicación del criterio profesional experimentado, a un problema recurrente, la idea de patrones de Alexander podrá usarse para reutilizar y transmitir dichas experiencias a profesionales menos experimentados.

## 2.5 Calidad en la Ingeniería de Software

Hablar de calidad es entrar a un concepto complejo, existen múltiples interpretaciones del término de diferentes autores e instituciones ( (Deming, 1989), (Crosby, 1996), (Juran & De Feo, 2010), (ISO-9000, 2017)). Garvin añade: “la calidad es un concepto complejo y de facetas múltiples” (Garvin, 1984).

- Punto de vista trascendental. Se percibe de inmediato, pero no es posible definirla.
- Punto de vista del usuario. Satisfacer las necesidades del cliente.
- Punto de vista del fabricante. Cumplir con las especificaciones preestablecidas del producto.
- Punto de vista del producto. La calidad tiene que ver con las características inherentes del producto: funciones y características.
- Punto de vista del valor. Cuando la calidad está medida acorde con lo que el cliente está dispuesto a pagar por el producto.

Entrando al ámbito de la ingeniería de software tenemos la definición de Pressman (Pressman, 2002): “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características que se espera de todo software desarrollado profesionalmente”.

Podemos notar acerca de la definición de calidad de Pressman, que se orienta tanto a las especificaciones del producto como a las características del software que satisfagan al cliente. En este sentido, el producto del proceso de software debe tener ciertas características como son la eficiencia, la flexibilidad, la corrección, la confiabilidad, la mantenibilidad, la portabilidad, la usabilidad, la seguridad y la integridad (Pressman & Maxim, 2015). A pesar de esto, que un software cuente con todas estas características no asegura que satisfaga los requisitos del cliente.

Hoy en día, la calidad del software se ha convertido en un tema vital en el campo de la ingeniería de software. Es por ello que muchos estudios se orientan al desarrollo y mejora de métodos y técnicas de calidad para la ingeniería de software. Por ejemplo, Kaur hace un estudio de modelos de calidad de productos de software y propone un mejorado (Kaur, et al., 2013). Por su parte, Koh propone un modelo basado en diferentes vistas o perspectivas del software para analizar su calidad (Koh, 2016).

En la ingeniería de requisitos, como parte de la ingeniería de software, la calidad dependerá de cuán bien responda el producto a las expectativas del cliente. Como el producto son los requisitos, la calidad dependerá de cuán bien la especificación de requisitos refleje lo que realmente necesita el cliente (el cliente del software propiamente dicho y los ingenieros de software). Por consiguiente, podemos definir la calidad de la especificación en ingeniería de requisitos como la diferencia entre requisitos buenos y malos desde el punto de vista de los ingenieros de software (específicamente diseñadores) y los clientes del software.

En consecuencia, existen muchos trabajos que se enfocan en mejorar la calidad de la ingeniería de requisitos. Trabajos orientados a mejorar el proceso de la ingeniería de requisitos usando diferentes técnicas y herramientas ( (Saeed, et al., 2016), (Sankhwar, et al., 2014), (Horkoff & Yu, 2016), (Shah & Patel, 2014)). Tenemos otros trabajos que proponen formatos de especificación, y cualidades y formas de organización de requisitos ( (CC, 2017), (ESA, 2017), (Metrica-3, 2017), (ISO/IEC/IEEE-29148, 2017), (Pressman & Maxim, 2015), (Génova, et al., 2017), (Sommerville, 2016), (Hurtado, et al., 2009), (Saavedra, et al., 2013)). Tenemos otros trabajos que inciden directamente en formas de mejorar las cualidades de calidad de los requisitos. Zapata (Zapata, et al., 2006) propone una técnica para mejorar la completitud de los requisitos. Génova plantean un conjunto indicadores y factores que permiten medir y controlar la calidad de diferentes activos de la ingeniería de requisitos. Saavedra hace un estudio de diferentes modelos de calidad para mejorar la ingeniería de requisitos (Saavedra, et al., 2013).

En MORORE definimos un conjunto de factores y métricas específicas que permiten medir la calidad de los activos de requisitos que serán almacenados para su reuso efectivo. En este sentido, el tema de calidad de los requisitos está relacionado con la presente tesis doctoral.

### **2.5.1 Importancia de la calidad de requisitos**

La ingeniería de requisitos representa la primera etapa de la vida de un software y determina lo más crítico: qué se va a producir. Muchas veces se le resta importancia a esta etapa, cuando debería ser todo lo contrario; pues se ha mencionado ( (Braude & Bernstein, 2016), (Brooks, 1987), (Standish, 2016), (SWEBOK, 2017)) que muchos de los fracasos de los proyectos de software se debe a una inadecuada especificación de los requisitos. Un error cometido en la etapa de requisitos genera, a corto o largo plazo, pérdidas cuantiosas, no sólo en el ámbito económico, sino también en el ámbito social, personal, etc.

A continuación, exponemos algunas razones acerca de la importancia de realizar una ingeniería de requisitos de calidad:



# ESTADO DE LA CUESTIÓN

---

- En primer lugar, determinar bien lo que se va a producir aumenta las probabilidades de éxito del proyecto. Pues el documento de requisitos representa el consenso entre el cliente y el ingeniero de software. Por un lado, los ingenieros tendrán una guía clara de lo que deben diseñar e implementar; y por otro lado los clientes saben qué esperar acerca del producto que se está desarrollando.
- En segundo lugar, la ingeniería de requisitos bien desarrollada permite una mejor organización y planificación de las actividades a realizar en etapas posteriores; y mejora la forma en que se distribuyen y orientan los recursos. Con esto se puede proyectar un costo más real del proyecto y aproximar la duración del mismo; evitando así retrasos y aumentos de costos innecesarios.
- Además, cabe recordar que reparar un error que se ha cometido en las etapas iniciales del desarrollo de un software, y no se ha descubierto a tiempo, resulta mucho más caro y difícil que un error cometido en etapas posteriores; ya que requiere de mayor tiempo y esfuerzo para solucionarlo, por parte del equipo de ingenieros.
- La ingeniería de requisito también es importante para medir la calidad del software completo, ya que la calidad del software consiste justamente en el cumplimiento de los requisitos, tal y como se han especificado. Si los requisitos están claros, se podrá probar efectivamente el software y comprobar si se ha cubierto las necesidades del cliente (funcionalidad, facilidad de uso, confiabilidad, etc.). O en su defecto identificar cuáles necesidades aún no están satisfechas.
- Finalmente, la ingeniería de requisitos permite evaluar la productividad de un equipo. Ya que los requisitos son el objetivo y la meta a alcanzar, los ingenieros de software pueden saber cuándo han conseguido la meta o cuán cerca están de llegar a ella. Y por tanto pueden medir cuán productivo es el equipo de desarrollo.

Por todo lo mencionado anteriormente, la ingeniería de requisitos debe desarrollarse con especial cuidado y atención; empleando el tiempo y los recursos adecuados. Además, se debe invertir en desarrollar técnicas y herramientas que faciliten y mejoren el trabajo realizado por el ingeniero de requisitos; contribuyendo así en un aumento considerable en el porcentaje de proyectos exitosos.

## 2.6 Estándares de requisitos

Como hemos sustentado en la introducción, la ingeniería de requisitos es una de las fases más críticas del proceso de desarrollo de software. En este sentido, las investigaciones se orientan al mejoramiento de esta disciplina. La creación de los estándares relacionados con la ingeniería de requisitos es una respuesta al mejoramiento de este proceso. A continuación, expondremos los principales estándares que fueron fuente importante de características e ideas para el desarrollo de MORORE.

### 2.6.1 Métrica. Versión 3

Es la metodología de planificación, desarrollo y mantenimiento de sistemas de información creada por el Ministerio de Administraciones Públicas de España (Metrica-3, 2017). Esta metodología es útil para la sistematización de las actividades que dan soporte al ciclo de vida del software y pretende alcanzar los siguientes objetivos:

- Definir sistemas de información.

## ESTADO DE LA CUESTIÓN

- Conseguir productos software de calidad dando mayor importancia al análisis de requisitos.
- Mejorar los sistemas de información para que puedan adaptarse mejor a los cambios y favoreciendo la reutilización en la medida de lo posible.
- Facilitar la comunicación entre los distintos grupos que están produciendo software en todo el ciclo de vida de este.
- Facilitar el uso y el mantenimiento del producto software obtenido.
- Asegurar que el proyecto software cumple sus objetivos en calidad, coste y plazos asignados a su producción.

Las actividades propuestas según el estándar de Métrica V3 pueden ser automatizadas y modeladas, es más, la mayoría de estas técnicas están soportadas por una amplia variedad de herramientas CASE que hay en el mercado relacionadas con la ingeniería del software. Métrica V3 pretende cubrir todas las actividades que darán soporte al ciclo de vida del software, abarca el desarrollo completo de sistemas de información independientemente de su complejidad y magnitud, y sea cual sea la metodología, estructurada u orientada a objetos.

El estándar descompone cada uno de los procesos en actividades, y éstas a su vez en tareas. Para cada tarea se describe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes. Métrica V3 no establece una secuencia de actividades para que deban realizarse en el tiempo una detrás de otra, sino que pueden hacerse en paralelo; sin embargo, para terminar un proceso deben terminar todas las actividades del mismo que hayan sido fijadas para ese proceso.

Como mencionamos Métrica V3 pretende cubrir todo el proceso de desarrollo de software, pero centraremos nuestra atención en el proceso de análisis de sistemas de información de la metodología. De este proceso obtendremos la forma que sugiere el estándar para la especificación de requisitos (atributos y cualidades de requisitos), las formas organizativas de los mismos y el proceso que sigue. Estas características son un referente importante para la definición de las propiedades análogas dentro de MORORE.

Métrica V3 sugiere una forma de organizar requisitos que consta de un sólo nivel jerárquico (Tabla 2). También apreciamos que estos grupos son disjuntos, un requisito no puede pertenecer a más de una clasificación. Otro punto importante es las relaciones que pueden existir entre requisitos de clasificaciones diferentes, por ejemplo, un requisito de rendimiento restringe a uno o más requisitos funcionales.

Tipos	Definición
Requisitos funcionales	Definen las funcionalidades que debe realizar el sistema o componente.
Requisitos de rendimiento	Definen límites a funciones del software. Especifica aspectos como: velocidad de operaciones, rangos de precisión, uso de memoria, etc.
Requisitos de seguridad	Definen los criterios de seguridad para el software. Especifica aspectos tales como restricciones de acceso, confidencialidad, integridad, etc.
Requisitos de implantación	Definen las características de entrega y puesta en marcha del software.
Requisitos de disponibilidad	Definen niveles de disponibilidad del software, cómo podrá usarse el sistema, su capacidad mínima y media disponible, etc.

*Tabla 2 Estructura de tipos de requisitos, estándar Métrica V3.*

## ESTADO DE LA CUESTIÓN

Destacamos también que la especificación de requisitos es uniforme en todas las clasificaciones, no se definen atributos específicos para cada uno de estos grupos. El estándar Métrica V3 propone 8 atributos de requisitos, aunque deja a elección de los ingenieros la adición o no de otros atributos complementarios que mejoren la calidad y comprensión de los requisitos. Estos atributos, mostrados en la Tabla 3, aparecen en un documento complementario del estándar denominado “Técnicas y Prácticas”.

Atributo	Descripción de Atributo
Identificador	Nombre del requisito, conciso y no ambiguo, que hace referencia a un único requisito del sistema.
Autor	Nombre del analista del sistema de información encargado de definir con el usuario y/o cliente un determinado requisito.
Tipo de requisito	Especifica la clasificación del requisito, puede ser de tipo funcional, de rendimiento, de seguridad, de implantación o de disponibilidad.
Descripción	Especificación detallada del requisito a fin de evitar posibles ambigüedades, o falta de información. Podría incluir menciones a los datos de entrada y/o de salida involucrados, interfaces con elementos externos.
Prioridad	Indica el orden temporal de realización del requisito. Puede tomar los valores de: alta, baja, media.
Estado	Especifica la situación del requisito durante el SDP. Puede tomar valores de: propuesto, aprobado o incorporado.
Fecha de creación	Indica la fecha en que se especificó un determinado requisito por vez primera.
Fecha de revisión	Indica la fecha en que se modificó un determinado requisito creado anteriormente.
Otros atributos	Algunos ejemplos posibles son: necesidad (si es un requisito esencial o no, protocolos usados, costos máximos, etc.).

*Tabla 3 Atributos de requisitos, estándar Métrica V3.*

Métrica V3 también sugiere algunas cualidades que deberían tener los requisitos (factores de calidad) para su correcta definición. Métrica V3 enuncia las cualidades mostradas en la Tabla 4, con el propósito de especificar requisitos de calidad y minimizar la posibilidad de inconvenientes en las otras fases del proceso de desarrollo de software.

Cualidad	Descripción
Conciso	Debe especificar claramente y en breves palabras el propósito del requisito.
No ambiguo	El requisito no debe tener más de una interpretación. Debe tener un único significado.
Consistente	Un requisito no debe contradecir a otro y debe evitar extenderse al ámbito de otros requisitos.
No duplicado	Un requisito no puede estar duplicado parcial o totalmente por otro de los requisitos que se encuentren ya especificados en el sistema.
Completo	El conjunto de requisitos de un catálogo tiene que cubrir toda la semántica a la que se refiere el sistema.
Válido	El requisito debe expresar con la mayor aproximación posible los deseos o necesidades de los usuarios.

*Tabla 4 Cualidades de requisitos, estándar Métrica V3.*

Según Métrica V3, el conjunto de requisitos que se especifiquen de acuerdo a los atributos descritos en la Tabla 3 y las cualidades descritas en la Tabla 4 darán lugar a un catálogo de requisitos de calidad.

### 2.6.2 Agencia Espacial Europea (ESA: European Space Agency)

Los estándares de ingeniería del software de ESA definen las prácticas de software que deben aplicarse en todos los proyectos de la Agencia Espacial Europea (ESA, 2017). Estos estándares se ocupan de todos los aspectos software de un sistema, incluyendo las interfaces con el hardware u otros componentes. Fue concebido para albergar tanto grandes proyectos como pequeños. También se definen documentos específicos para el desarrollo de proyectos con métodos orientados a objetos.

ESA sugiere la determinación de factores previos tales como: el costo de desarrollo, la cantidad de personas implicadas, la cantidad de software que se producirá, etc. Una vez determinado el alcance del proyecto, hay que seguir el ciclo de vida software marcado por el estándar. Seis fases componen el ciclo completo:

- RU-Definición de Requisitos de Usuario
- RS-Definición de Requisitos Software
- DA-Definición del Diseño Arquitectónico
- DD-Diseño Detallado y Producción del Código
- TR-Transferencia del Software a las Operaciones
- OM-Operaciones y Mantenimiento

Asimismo, ESA incluye una serie de estrategias que facilitan el desarrollo del ciclo de vida del software para proyectos pequeños. Las estrategias incluyen la unión de ciertas fases, simplificación de la documentación, simplificación de los planes de todo el proceso, reducción en la exhaustividad de ciertos procesos, etc.

En cuanto al tema principal de la tesis doctoral, la ingeniería de requisitos, ESA define dos fases distintas pero muy relacionadas: definición de requisitos de usuario (RU) y definición de requisitos software (RS). Cabe destacar a su vez, que en estas fases el estándar ESA aplica una estructura de tipos compuesta de una serie de subdivisiones o clasificaciones de los requisitos según sea la naturaleza de los mismos. Así mostramos en la Tabla 5 que los requisitos de usuario se dividen en requisitos de capacidad y requisitos de restricción, mientras que los requisitos software se dividen en requisitos funcionales, requisitos de interfaz, requisitos de rendimiento, etc.

Como hemos mencionado, un correcto análisis y descripción de los requisitos (tanto software como de usuario) asegurarán el éxito de un proyecto software. En consecuencia, aparte de la estructura de tipos, ESA define un conjunto de atributos y cualidades que debería tener una adecuada especificación de requisitos.

## ESTADO DE LA CUESTIÓN

Tipo	Subtipo	Descripción
Definición de requisitos de usuario	Requisitos de capacidad	Definen las funciones del software según el punto de vista del usuario.
	Requisitos de restricción	Establecen limitaciones a los requisitos de capacidad.
Definición de requisitos de software	Requisitos funcionales	Definen una función característica del futuro software.
	Requisitos de rendimiento	Especifican valores numéricos para medir variables relacionadas con una función. Por ejemplo: velocidad, frecuencia, rapidez, capacidad y precisión.
	Requisitos de interfaz	Especifican los elementos hardware, software o bases de datos con los que el sistema se comunicará.
	Requisitos operacionales	Especifican cómo funcionará el software y cómo se comunicará con los operadores humanos (diseño de la pantalla, estilo del lenguaje de comandos, etc.).
	Requisitos de recursos	Especifican los límites de recursos físicos. Por ejemplo: la potencia de procesamiento, memoria principal o espacio en disco, etc.
	Requisitos de verificación	Definen características que facilitan la verificación de los requisitos.
	Requisitos de pruebas de aceptación	Especifican requisitos para la fase de transferencia de software a las operaciones.
	Requisitos de documentación	Definen el formato y estilo de la documentación.
	Requisitos de seguridad	Definen características para asegurar el software frente amenazas externas de confidencialidad, integridad y disponibilidad.
	Requisitos de portabilidad	Especifican la facilidad de cambio del software de un entorno a otro.
	Requisitos de calidad	Definen características que hacen al software apropiado para su propósito.
	Requisitos de fiabilidad	Especifican el tiempo medio aceptable entre dos fallos producidos en el software.
	Requisitos de mantenimiento	Definen la facilidad con que el software deberá ser modificado para corregir defectos, mejorar su desempeño, etc.
	Requisitos de protección	Definen la seguridad para personas frente a fallos del sistema.

*Tabla 5 Estructura de tipos de requisitos, estándar ESA*

Los atributos describen aquellos aspectos que deberán definir la totalidad de cada requisito, es decir, el conjunto de todos los atributos, dan forma a cada requisito. Describir de manera concreta y completa los requisitos para cada proyecto será de vital importancia. En este sentido ESA propone en forma concisa los atributos que mostramos en la Tabla 6, sin negar la posibilidad que el ingeniero añada otros atributos a los requisitos en un proyecto específico.

Tan importante como describir todos los atributos necesarios de un requisito, es qué estos atributos cumplan una serie de cualidades. Mediante estas cualidades se asegura que las especificaciones de requisitos tengan los mínimos de calidad esperados. En este sentido, ESA propone que los requisitos cumplan las cualidades descritas en la Tabla 7. Estas cualidades de los requisitos son las más representativas para el estándar ESA, y su propósito es que la especificación de requisitos tenga unos mínimos niveles de calidad.

## ESTADO DE LA CUESTIÓN

Atributo	Descripción
Identificador	Nombre del requisito, claro y conciso, que facilite su trazabilidad a través las siguientes fases.
Necesidad	Aquellos requisitos marcados como “Esenciales” son de vital importancia y no están sujetos a negociaciones.
Prioridad	Cada requisito debe incluir una medida de prioridad a fin de que el desarrollador pueda decidir el calendario de producción y plazos de entrega.
Estabilidad	Se definen aquellos requisitos que deben ser estables durante todo el ciclo de vida del proyecto y aquellos que pueden sufrir cambios debido al diseño arquitectónico o al diseño detallado y producción del código.
Origen	Hace referencia al documento externo, usuario o grupo de usuarios que establecen un determinado requisito.
Claridad	Se incluirán especificaciones detalladas y descripciones a fin de evitar cualquier tipo de ambigüedad.
Verificabilidad	Se incluirán comprobaciones para que el requisito sea incorporado a las fases diseño, así como implementado y probado.

*Tabla 6 Atributos de requisitos, estándar ESA.*

El estándar ESA no diferencia las cualidades según los tipos de requisitos, se considera que todos los requisitos deban cumplir con estas cualidades.

Cualidad	Descripción
Trazable	Que el requisito sea trazable, que desde el origen del requisito podamos recorrerlo hasta el fin de este requisito en la implementación.
Unívoco	El requisito sólo se referencia a sí mismo.
Claro	El requisito debe de ser claro en su explicación y la parte a la que se refiere él.
Conciso	Descripción breve y define exactamente lo referido al requisito sin salirse del contexto de éste.
No ambiguo	No puede existir ambigüedad en el requisito ni en su definición, así que habrá que tener cuidado con el lenguaje utilizado para definirlo.
Verificable	El requisito debe de ser verificable, por algún medio demostrar que ese requisito puede ser probado.

*Tabla 7 Cualidades de requisitos, estándar ESA.*

### 2.6.3 ISO/IEC 29148:2011

El estándar ISO/IEC/IEEE 29148:2011 (ISO/IEC/IEEE-29148, 2017) describe las prácticas recomendadas para la especificación de requisitos software, definen los contenidos del documento y las cualidades que debe tener una buena descripción de requisitos software. En este sentido, el estándar propone una serie de orientaciones generales y específicas que debemos tener en cuenta en el proceso y definición de los requisitos de software.

Orientándonos a nuestra investigación, el estándar ISO/IEC/IEEE 29148:2011 define una estructura de tipos para los requisitos que permite una eficiente clasificación de los mismos. Esta estructura, mostrada en la Tabla 8, tiene el propósito de cubrir todos los aspectos pertinentes para la descripción de requisitos del futuro software. En consecuencia, el estándar facilitará al ingeniero la especificación y gestión de los requisitos del futuro software. Esta estructura de tipos consta de dos niveles de organización: tipos y subtipos.

## ESTADO DE LA CUESTIÓN

Tipo	Subtipo
Requisitos de interfaces	Interfaces del sistema
	Interfaces de usuario
	Interfaces de hardware
	Interfaces de software
	Interfaces de comunicaciones
Requisitos funcionales	
Requisitos de rendimiento	Número de terminales a ser soportadas
	Número de usuarios simultáneos a ser soportados
	Cantidad y tipo de información a manejar
Requisitos de lógica de la base de datos	Tipos de información por funciones
	Frecuencia de uso
	Acceso a las capacidades
	Entidades de datos y relaciones
	Restricciones de integridad
	Retención de datos
Requisitos de cumplimiento de estándares	Formato de reportes
	Nombre de datos
	Procedimientos de contabilidad
	Lineamientos de auditoria
Requisitos de cualidades del software	Fiabilidad
	Disponibilidad
	Security
	Mantenibilidad
	Portabilidad

*Tabla 8 Estructura de tipos de requisitos, estándar ISO/IEC/IEEE 29148.*

El estándar no define directamente los atributos de especificación de requisitos, es más sugiere que los atributos se definan en base al proyecto específico que se esté desarrollando. Pero después de la lectura del estándar, identificamos algunos atributos aplicables a cualquier requisito que tendrá el futuro software. La Tabla 9 muestra los atributos aplicables a cualquier tipo de requisitos según el estándar ISO/IEC/IEEE 29148:2011.

Atributo	Descripción
Identificación	Especificar el nombre conciso del requisito que lo haga unívoco.
Descripción	Especificar los objetivos perseguidos por el requisito.
Grado de estabilidad	Posibilidad de no sufrir cambios en el requisito durante el desarrollo del software.
Grado de necesidad	Especificar la importancia del requisito respecto al software.
Trazabilidad	Especificación de las relaciones de origen del requisito y también del destino del requisito.

*Tabla 9 Atributos comunes de requisitos, estándar ISO/IEC/IEEE 29148:2011.*

En cuanto a las cualidades que debería tener un requisito, el estándar ISO/IEC/IEEE 29148:2011 define un conjunto de características que debería tener una buena especificación de requisitos. Como vemos en la Tabla 10, especificamos las cualidades descritas en el estándar. Según el estándar, los requisitos deberán cumplir con estas características para lograr una especificación de calidad dentro un proyecto de desarrollo de software.

## ESTADO DE LA CUESTIÓN

Cualidad	Descripción
Correcto	Debe reflejar las necesidades o deseos reales del cliente
No ambiguo	Debe tener una sola interpretación
Completo	Debe contemplar toda la semántica que se refiere a ese requisito, y en forma global cubrir todos los aspectos del software
Consistente	Debe evitar los conflictos entre los requisitos
Clasificables (importancia y/o estabilidad)	Debe permitir ser clasificado por la importancia y la estabilidad
Comprobable	Debe definir un objetivo que permita su fácil comprobación
Modificable	Debe tener una estructura y estilo que permita hacerle cualquier cambio al requisito fácilmente
Identificable	Se debe definir claramente el origen del requisito y facilitar la referencia con respecto a elementos del diseño e implementación.

Tabla 10 Cualidades de requisitos, estándar ISO/IEC/IEEE 29148:2011.

### 2.6.4 Common Criteria for Information Technology Security Evaluation (CC)

Existían grandes discrepancias entre las organizaciones sobre cómo definir la seguridad en las tecnologías de información. Los primeros esfuerzos de unificación al respecto surgen a principios de los 80, en los Estados Unidos, donde se desarrollaron los criterios de seguridad bajo el nombre TCSEC (Trusted Computer System Evaluation Criteria) y conocidos como *Orange Book* (Orange-Book, 1985). Estos criterios sirvieron como base para que otros organismos internacionales los evolucionaran haciéndolos más flexibles y adaptables al continuo desarrollo de las tecnologías de la información.

Ya en la década de los 90 la ISO (International Organization for Standardization) empieza a trabajar en la necesidad de estandarizar internacionalmente los requisitos de seguridad para las tecnologías de la información. En 1996 aparece el CC versión 1.0 y llega a consolidarse ya en 1999 con la versión 2.1 o ISO-IEC 15408. En la actualidad, después de numerosas revisiones, contamos con la versión 3.1 en su revisión 4 (CC, 2017).

El estándar CC o ISO/IEC 15408 (International Organization for Standardization / International Electrotechnical Commission), se orienta a la evaluación de la seguridad de las tecnologías de la información. Esta norma es cada vez más requerida y utilizada a nivel mundial. Pero su gran extensión y alcance hacen que la adaptación de esta norma a proyectos específicos no sea sencilla, sobre todo desde el punto de vista de los ingenieros (Fernández-Medina, 2003).

Con el grado de complejidad actual de los sistemas de información, obtener con certeza sus propiedades de seguridad es un problema de difícil solución. Ante esta situación el CC basa su filosofía, en concordancia con el estado del arte del análisis de seguridad, en ofrecer grados de confianza en la seguridad de un producto o sistema de información.

El estándar está organizado en tres partes con distintas orientaciones, pero relacionadas entre sí. En primer lugar, se tiene una introducción al modelo general con un conjunto de definiciones y principios de evaluación de la seguridad. En segundo lugar, se presentan un conjunto de requisitos de seguridad llamados componentes funcionales de seguridad (Security functional components) para hacer frente a las amenazas del producto. Y en tercer término presenta un conjunto de requisitos de garantía de la



## ESTADO DE LA CUESTIÓN

---

seguridad llamados componentes de garantía de la seguridad (Security assurance components).

El CC permite obtener un grado de confianza en su seguridad a todo tipo de producto o sistema en los que se necesite. Tiene una organización de sus criterios de seguridad muy estructurada, aunque a la vez muy flexible. Aun así, visto desde un nivel de organizaciones de desarrollo de productos o sistemas de información, no es sencillo de aplicar.

Todos los requisitos de seguridad del CC, tanto los de funcionalidad como los de garantía, se organizan de forma jerárquica en clases, familias y componentes. Estas agrupaciones de criterios de seguridad son de reconocida validez y permiten a los usuarios la fácil localización de los requisitos, para su posterior uso.

En la Figura 5 mostramos la estructura de requisitos del estándar CC, la cual consta de los elementos descritos a continuación:

- Clase. Nivel más genérico de agrupamiento de los requisitos de seguridad. Compuesto por familias, que comparten un mismo enfoque, aunque se diferencian en el alcance de sus objetivos de seguridad.
- Familia. Compuesta por componentes, los cuales comparten objetivos de seguridad, pero pueden diferir en el énfasis o rigor. Los componentes dentro de una familia pueden ser jerárquicos, lo cual permite a un componente cualquiera satisfacer dependencias de otro dentro de la jerarquía.
- Componente. Define un conjunto específico de requisitos de seguridad. Es la parte “seleccionable” más pequeña para su uso en la estructura del CC. Los componentes son construcciones de elementos individuales. El elemento, es la expresión más pequeña de requisito de seguridad, y es el requisito indivisible de seguridad que puede ser verificado para la evaluación.

Los otros elementos de la Figura 5: paquetes, perfil de protección y declaración de seguridad, son conjunto de especificaciones y de componentes creados para el uso de los requisitos de seguridad del Common Criteria. Especificamos sus características más adelante.

- Declaración de seguridad. Define un conjunto de especificaciones y de requisitos de seguridad de un producto informático específico (objetivo de evaluación). Esta declaración de seguridad define los límites y detalles específicos del objeto de evaluación. Es ofrecida por el proveedor como base para la evaluación.

## ESTADO DE LA CUESTIÓN

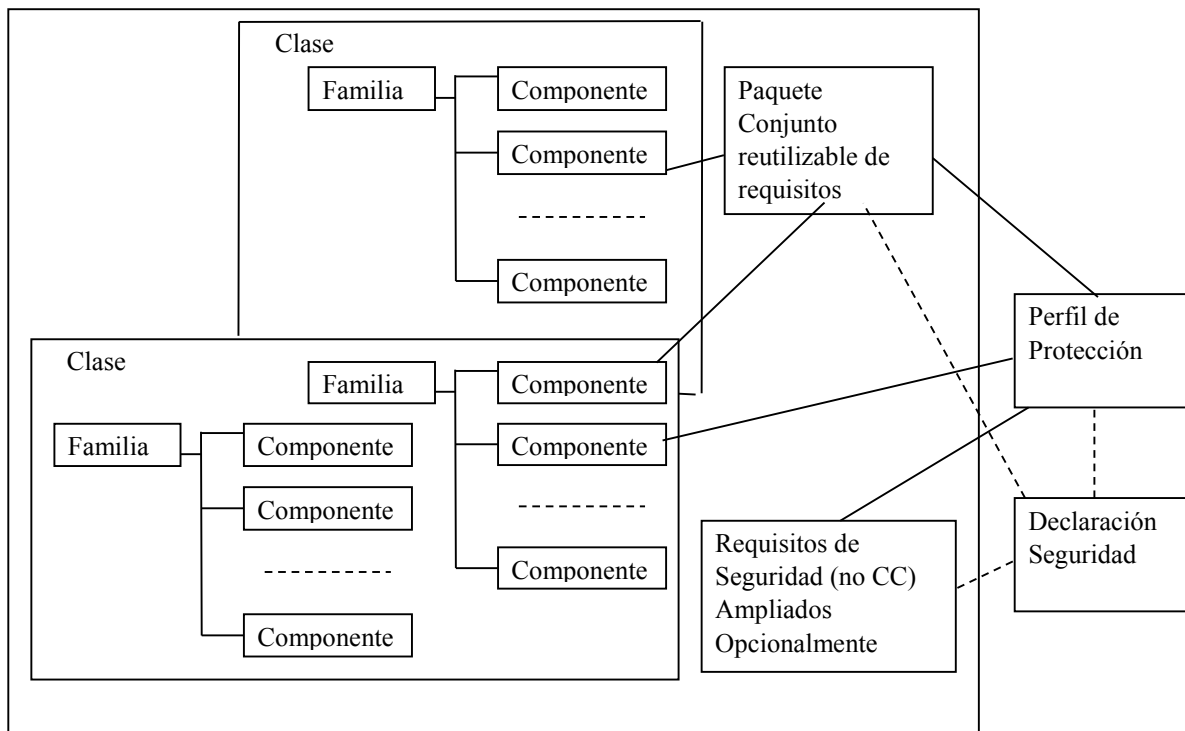


Figura 5 Organización de requisitos de seguridad, estándar CC.

Nomenclatura de los requisitos.- Mediante el ejemplo de la Tabla 11, mostramos la nomenclatura usada para los requisitos de seguridad en el CC.

Nombre del requisito	FAU_GEN.1.2
F	Requisito funcional (“A” para los de garantía).
AU	Clase Auditoría de Seguridad.
_GEN	Familia Generación de datos.
.1	Primer componente, llamado “Generación de datos de auditoría”.
.2	Segundo elemento del componente.

Tabla 11 Ejemplo de la nomenclatura de un requisito en el CC.

Dependencias entre los componentes. Se dan cuando el componente no es autosuficiente y depende de otro para su uso. Las dependencias deben ser satisfechas para usar el componente en un Perfil de Protección (PP: Protection Profile) o Declaración de Seguridad (ST: Security Target).

Operaciones permitidas en componentes. Los componentes pueden ser usados tal como están definidos en el CC, o ser adaptados a través del uso de operaciones permitidas, para alcanzar una política de seguridad específica o contrarrestar una amenaza determinada. Estas operaciones son: iteración, asignación, selección y refinamiento.

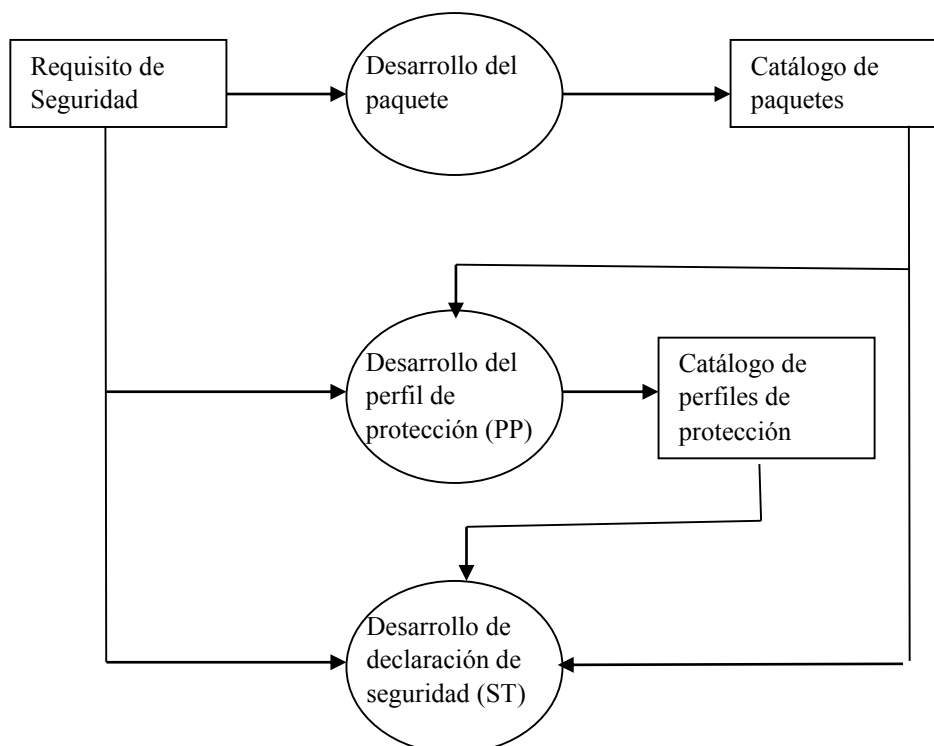
## ESTADO DE LA CUESTIÓN

Para el uso de los requisitos de seguridad del CC, se definen tres tipos de estructuras de requisitos: Paquete (Package), Perfil de Protección (PP) y Declaración de Seguridad (ST), los cuales los mostramos en la Figura 6 y describimos a continuación.

**Paquete (Package).** Permite la expresión de un conjunto de requisitos funcionales o de garantía (componentes de distintas clases o familias) que cumplen un subconjunto identificable de objetivos de seguridad. Un paquete está pensado para ser reutilizable y puede ser usado en la construcción de paquetes mayores, PPs y STs.

**Perfil de Protección (PP).** Define un conjunto de requisitos de seguridad (componentes o paquetes), independientes de la implementación, para un tipo o categoría de software (objeto de evaluación) que cumplirán un conjunto de objetivos de seguridad comunes a varios usuarios.

**Declaración de Seguridad (ST).** Define un conjunto de requisitos de seguridad específicos para un determinado software (objeto de evaluación), como base para su posterior evaluación. Contiene la especificación resumen de un producto específico, los requisitos y objetivos de seguridad, y el fundamento de cada uno. Pueden basarse en un PP.



*Figura 6 Uso de requisitos de seguridad, estándar CC.*

Para el desarrollo de un Paquete se requiere la composición de un conjunto de requisitos de seguridad. El Paquete desarrollado pasa a conformar un catálogo de paquetes. Para el desarrollo de un Perfil de Protección (PP) se requiere igualmente un conjunto de requisitos de seguridad o de Paquetes ya creados. El PP desarrollado pasa a conformar un catálogo de PPs. Y para el desarrollo de una Declaración de Seguridad (ST) se puede requerir la composición tanto de requisitos de seguridad como de PPs y STs.

### 2.7 Proceso: PMBoK

El Project Management Institute (PMI) es una organización sin fines de lucro. Esta organización promueve la profesión de la dirección de proyectos a través de estándares y certificaciones reconocidas mundialmente, a través de comunidades de colaboración, de un extenso programa de investigación y de oportunidades de desarrollo profesional (PMI, 2017).

Uno de los productos más importantes del PMI es el PMBoK (the Project Management Body of Knowledge) (PMBoK, 2013). El PMBoK es una guía de procesos estandarizados para que los profesionales puedan aplicar dichos procesos a cada caso o situación particular. Los procesos descritos en el PMBoK son reconocidos como buenas prácticas que se pueden aplicar a la mayoría de los proyectos por parte de los involucrados.

Entre otros puntos, en el PMBoK, están descritos un conjunto de 47 procesos, divididos en 10 áreas de conocimiento y 5 grupos de procesos. Cada proceso está descrito con los siguientes elementos:

- Proceso: representa específicamente la buena práctica a seguir en el proyecto.
- Entradas: elementos que proveen lo necesario para realizar el proceso.
- Salidas: productos específicos que pueden servir como entrada a otros procesos.
- Herramientas y técnicas: facilitan la realización del proceso.

En MORORE planteamos de forma complementaria una guía de pasos a seguir por el ingeniero para el almacenamiento y reúso de los activos de requisitos. Este conjunto de pasos, los especificamos en base a la forma de descripción de procesos del PMBoK (proceso, entradas, salidas, herramientas y técnicas) y añadimos el elemento: criterios de validación. De esta manera, el PMBoK del PMI tiene relación con la presente tesis doctoral.

### 2.8 Herramientas CASE: gestión de requisitos

El desarrollo de grandes y complejos sistemas de información presenta muchas dificultades para los ingenieros software. La principal dificultad es la capacidad de asegurar que el sistema satisfaga las necesidades de los usuarios finales. Estas necesidades o deseos normalmente cambian y sufren modificaciones a lo largo del ciclo de vida del software. Estos cambios hacen difícil tener un seguimiento de la evolución del software respecto a los requisitos iniciales proporcionados por el usuario. En este sentido es necesario realizar una buena especificación de los requisitos que permita una eficiente gestión de los mismos.

La clave del éxito de cualquier proceso de gestión de requisitos es la trazabilidad. La trazabilidad de requisitos es una técnica que se utiliza para proporcionar relaciones entre requisitos, elementos de diseño y elementos implementación de un software. Esta técnica permite poder gestionar los cambios y asegurar el éxito de los sistemas entregados.

Para que la especificación y gestión de requisitos sea eficiente, las diferentes técnicas usadas deben estar soportadas por herramientas informáticas. Por tanto, estas herramientas deben servir de apoyo a todo el proceso de desarrollo de software. En el presente estudio se propone que MORORE esté soportado por una herramienta automática de gestión de requisitos, que permita fehacientemente el reuso efectivo y eficiente de requisitos, motivo por el cual está relacionada la investigación con este tipo de aplicaciones.

### **2.8.1 Evolución de herramientas de gestión de requisitos**

En la década de los setenta, los ingenieros software utilizaban papel y lápiz para realizar seguimientos de los requisitos y de elementos del diseño de grandes y complejos sistemas de información. Estas elementales herramientas se utilizaban para diseñar y desarrollar sistemas complejos de comunicación, transporte, aeroespaciales y militares. Aunque estas herramientas fueron las mejores en su tiempo, la gestión de centenares de requisitos complejos y la toma de decisiones consumía mucho tiempo de los ingenieros, dejando pocas oportunidades para explorar alternativas o realizar análisis.

Durante los años ochenta, los ingenieros utilizaban procesadores de texto, hojas de cálculo, bases de datos, etc. como herramientas para diseñar y desarrollar sistemas. Estas herramientas proporcionan una gran mejora con respecto a las herramientas anteriores, pero siguen marcando limitaciones para los ingenieros. Hasta hace muy poco no era posible transferir datos de una herramienta a otra, o de un entorno a otro. Las herramientas no soportaban la capacidad de visualizar la información de los requisitos en cualquier nivel de abstracción. La mayoría eran diseñadas para un solo usuario y no era fácil hacer una transición para soportar entornos de desarrollo de sistemas grandes y con varios equipos.

Hoy en día muchas herramientas automatizadas y especializadas de ingeniería de software están disponibles. La creación de estas herramientas ha sido posible gracias a las grandes mejoras de la tecnología informática: la capacidad de almacenamiento y la velocidad de procesamiento. Con estas mejoras, la tecnología informática es capaz de soportar procesos complejos necesarios para la automatización del proceso de la ingeniería de software.

Entre las herramientas automatizadas que han surgido existen las herramientas especializadas en la gestión de requisitos. Estas herramientas se concentran en la captura, especificación y gestión de requisitos. Aun así, estas herramientas tienen limitaciones, que son motivo de estudio de muchos investigadores.

MORORE se concibe como un modelo que permite el reuso de activos del ámbito de la ingeniería de requisitos. Como lo mencionamos, la aplicación de este modelo en una herramienta de gestión de requisitos permitirá aprovechar al máximo las potencialidades descritas. Tomar en cuenta las características de las herramientas para la gestión de requisitos permitió dar a MORORE la orientación debida para su futura implementación.

### **2.8.2 Características generales de las herramientas de gestión de requisitos**

Las herramientas disponibles hoy en día están, principalmente, enfocadas a la trazabilidad y la organización (con algunas excepciones). Las herramientas varían según

el nivel de soporte que brindan a las actividades de la gestión de requisitos. La mayoría de las herramientas proporcionan vistas de la gestión de requisitos orientadas al texto (por ejemplo: DOORS (DOORS, 2017), RTM (SERENA, 2017), y otras). Pocas herramientas tienen vistas de la gestión de requisitos orientadas a modelos (como por ejemplo: CORE (CORE, 2017) y RDD-100 (RDD-100, 2017)).

El termino trazabilidad se utiliza para indicar una relación entre un requisito y cualquier otro elemento dentro del proceso de ingeniería de software, por ejemplo, un componente de diseño o un documento de especificación. Muchas herramientas soportan enlaces de trazabilidad entre requisitos y elementos de diseño, o entre requisitos y documentos relacionados.

Entre las capacidades organizativas de las herramientas resalta el agrupamiento de los requisitos en diferentes formas organizativas que por lo general están asociadas a un estándar de especificación de requisitos. Estas formas organizativas y los atributos de especificación que permiten las herramientas ayudan al proceso de gestión de requisitos.

Otras características disponibles en las herramientas de gestión de requisitos son la generación de informes a medida y la provisión de interfaces con otras aplicaciones y herramientas de la ingeniería del software. Asimismo, muchas herramientas de gestión de requisitos proporcionan interfaces de usuario con múltiples características funcionales que facilitan el trabajo de los ingenieros.

### **2.8.3 La trazabilidad como parte de la gestión de requisitos**

La trazabilidad se ha convertido en un elemento controlador clave para el proceso de desarrollo de software. Este elemento proporciona relaciones entre requisitos, elementos de diseño y elementos de implementación del software. La trazabilidad permite el mantenimiento y proporciona la capacidad de descubrir la historia de cualquier característica del proceso de desarrollo de software.

Existen muchas relaciones entre requisitos, elementos de diseño, componentes, otros activos y actividades del proceso de ingeniería de software. La gestión de estas relaciones es crítica para proporcionar una buena comprensión de la gestión de requisitos dentro del ciclo de vida del software. En este sentido, si la trazabilidad se utiliza de una manera consistente se podrá conocer lo siguiente:

- El impacto de la modificación de un requisito.
- Cómo se implementa un requisito.
- Si todos los requisitos están en proceso de desarrollo.
- La razón de ser del requisito.
- La importancia del requisito respecto a todo el software.
- El ámbito de requisito respecto a las decisiones de diseño e implementación.
- Si el diseño y la implementación es conforme al requisito.
- Las pruebas de aceptación que se van a utilizar con el requisito.

- Cuáles elementos de diseño son necesarios.

### 2.8.4 Herramientas en estudio

Para la presente tesis doctoral estudiamos herramientas automáticas de gestión de requisitos relevantes en el mercado. A continuación, mostramos las características relevantes de estas herramientas que han servido de base para la concepción del modelo motivo de la presente investigación: MORORE.

**Rational Requisite Pro** (Reppro, 2017). Herramienta que permite administrar requisitos. Esta herramienta permite especificar los requisitos, controla una base de datos que permite el almacenamiento y recuperación de los requisitos y permite soportar la trazabilidad entre tipos de requisitos y otros elementos del proceso.

Requisite Pro es parte de la solución Rational para la gestión de los requisitos. Se complementa con Rational Rose (Análisis y Diseño), Rational Soda (Documentación de Modelo) y Rational Suite TestStudio (Herramientas para la Automatización de las pruebas de Software), Rational ClearCase (Herramienta para el control y administración de versiones), y Rational Unified Process (Asistente de Metodología y Procedimientos de Desarrollo de Software).

**Telelogic DOORS** (DOORS, 2017). Es una suite para la administración de requisitos. Se orienta al mejoramiento de la calidad de la comunicación y colaboración entre los miembros del equipo de desarrollo de software. Consta de las siguientes características:

- Interfaces intuitivas que promueven el uso por parte de los ingenieros.
- Escalabilidad para cualquier tamaño de proyecto con cualquier número de usuarios.
- Soporte completo para el registro, estructuración, gestión y análisis de requisitos.
- Integración otras soluciones de Telelogic y herramientas de terceros para tener una mejor visión de los requisitos y controlar su trazabilidad a lo largo del ciclo de vida del proceso de desarrollo de software.

En este sentido, DOORS consta de tres entornos con diferentes funcionalidades, pero complementarias para el logro de sus objetivos:

- DOORS, entorno de formularios. Orientado a bases de datos que permite a los usuarios capturar, vincular, llevar el rastro, analizar y manejar la información de todos los requisitos de un proyecto, de tal forma que asegure que el proyecto cumpla con los requisitos iniciales y los estándares. Tiene el poder y la escalabilidad para manejar proyectos grandes, complejos y con muchos usuarios concurrentes que trabajan sobre una red.
- DOORSnet. Permite al equipo de trabajo acceder a los datos de los requisitos, modificarlos, consultarlos y planear cambios, usando un browser estándar a través de Internet, o de la intranet corporativa o desde una red LAN.
- DOORSrequireIT. Entorno basado en Microsoft Word para usuarios no técnicos que están acostumbrados a trabajar con este procesador de textos, facilitando la participación en un proyecto.

**IRQA** (IRQA, 2017). Es una herramienta CASE para ingeniería de requisitos, especialmente diseñada para soportar las actividades realizadas en el proceso de especificación de software. Permite dar soporte a los ingenieros en las actividades siguientes:

- Captura de requisitos.
- Especificación de requisitos.
- Gestión de requisitos.
- Organización y clasificación de los requisitos.
- Especificación de la solución.
- Gestión de pruebas de aceptación.

### **2.8.5 Características principales de las herramientas de gestión de requisitos**

Después del estudio de las herramientas (Rational Requisite Pro, Telelogic y IRQA) hemos seleccionado las funcionalidades que consideramos más relevantes para una herramienta de gestión de requisitos. A estas funcionalidades las clasificamos en categorías lógicas relativas a las tareas que desempeñan respecto al usuario. A continuación, mostramos estas funcionalidades:

- Captura e identificación de requisitos:
  - Variedad de documentos de entrada: permite importar documentos en diferentes formatos estándares, por ejemplo: formato texto (\*.txt), formato de texto enriquecido (\*.rtf), formato de Microsoft Project (\*.mpx), etc. El documento importado podría ser enriquecido editándolo con diferentes fines: corregir ortografía, editar el título o la descripción del requisito, agregar objetos en diferentes niveles de jerarquía, eliminar un requisito, agregar una tabla, etc.
  - Análisis de comparación y cambios en el documento de entrada: permite mostrar las modificaciones de los requisitos y del módulo del proyecto activo. Muestra el usuario, el número de sesión, la fecha, hora y la descripción de la modificación realizada. En resumen, muestra información sobre las diferentes versiones de los requisitos y del módulo en general.
  - Especificación de requisitos (interactiva): permite especificar las características de los requisitos. Entre otras las características son el título del requisito, un nombre corto, una descripción del requisito, etc.
- Estructura de los elementos
  - Estructura gráfica: permite visualizar los elementos de un documento en una estructura gráfica y jerárquica, mediante la característica de nombre corto.
  - Estructura textual: permite visualizar los requisitos de forma textual. Muestra sus características y su estructura mediante un sistema de numeración.



# ESTADO DE LA CUESTIÓN

---

- Flujo de los requisitos
  - Asignación de los requisitos a elementos del sistema: permite relacionar los requisitos con otros requisitos y otros elementos o activos del proceso de desarrollo de software (bidireccionalmente si es necesario). También debe permitir la asociación de los requisitos con otros elementos no pertenecientes al sistema, como por ejemplo archivos relacionados, direcciones web, etc.
  - Control de la configuración: permite llevar un control de los cambios producidos en los requisitos. Controla las diferentes versiones, problemas, estados, etc., de los requisitos.
  - Comparación de versiones: permite visualizar los cambios realizados entre distintas versiones de los requisitos.
- Análisis de Trazabilidad
  - Identificación de inconsistencias: permite mostrar los requisitos que están relacionados y los que no están relacionados (requisitos huérfanos).
  - Visibilidad de enlaces: permite visualizar los requisitos con sus respectivos enlaces o relaciones con otros requisitos y con otros elementos del sistema.
- Documentos y otros medios de salida
  - Especificación estándar de salida: permite exportar el documento de requisitos en varios formatos estándar: texto, Microsoft Office, HTML, XML, etc.
  - Diccionario de datos: permite la especificación de los principales conceptos del ámbito controlado por los requisitos.
  - Características adaptadas de salida: permite adaptar las características del documento para su salida. Por ejemplo: en texto simple, tablas, gráficos, y otras características que se pueden combinar según las necesidades.
- Trabajo en grupo
  - Soporta revisiones concurrentes: permite controlar el acceso para la realización de revisiones y comentarios a los documentos en un entorno multiusuario.
  - Control de asignación de permisos a usuarios: permite asignar a los usuarios los derechos respectivos para que manejen un conjunto de requisitos.
- Interfaces con otras herramientas
  - Comunicación con otras herramientas: permite la comunicación con otras herramientas mediante la exportación e importación de documentos en formatos estándares: formato texto, Microsoft Office, HTML, XML, etc. Asimismo, interacción con otros programas mediante interfaces para el uso de funciones y procedimientos externos, por ejemplo, uso de APIs (Interfaz de Programación de Aplicaciones) para medir la calidad de los requisitos.
  - Bases de datos: soporta sistemas abiertos de bases de datos (acceso de consultas estándar), que permita la importación directa de información de formatos estándares de base de datos.

### 2.9 Investigaciones relacionadas al reúso de requisitos

En esta sección mostramos los estudios de otros autores que demuestran el mismo interés de la comunidad científica por la finalidad que persigue la presente investigación: el mejoramiento de la ingeniería de requisitos por medio de técnicas de reúso.

#### 2.9.1 Patrones de análisis (Fowler, 1997)

Martín Fowler, ingeniero de software, escritor, orador y consultor acerca de desarrollo de software para empresas (Fowler, 2017). Es uno de los primeros en llevar la idea de patrones al ámbito de la ingeniería de requisitos. Propone un conjunto de modelos generales de requisitos (a los que llama “patrones de análisis”) de los diferentes aspectos de la lógica de negocio de una organización. Fowler, menciona que estos patrones pueden ayudarnos a simplificar el desarrollo de un sistema informático.

La propuesta de Fowler se basa en que el modelado conceptual debería ser independiente de la tecnología de software. Esta independencia tiene el propósito de evitar que la tecnología influya en la comprensión del problema y que el resultado sea útil en cualquier tecnología. En este sentido, los patrones de análisis reflejan la estructura conceptual (modelo) del proceso de los negocios, más que de la implementación del software.

En el análisis de sistemas existe la necesidad de modelar los conceptos relevantes de la aplicación que se va a construir. Los diagramas UML ayudan en gran medida a esta labor de modelamiento. Pero en la práctica los ingenieros muchas veces no tienen la experiencia ni referencias suficientes para realizar el modelado.

Los patrones de análisis de Fowler brindan modelos conceptuales generales de negocio realizados por expertos. Estos modelos abstraen, a un aspecto general, los conceptos y relaciones de una situación particular. Estos modelos reflejan las estructuras conceptuales de los procesos de negocio, en lugar que las aplicaciones de software reales.

Por ejemplo, para la comprensión de estos patrones, se puede partir de la situación particular de una tienda de venta de autos. En este análisis se tienen los siguientes conceptos: auto, cliente (comprador del auto), vendedor del auto y la propia venta del auto. Esto se puede generalizar respectivamente a: producto, cliente, vendedor y venta. Gráficamente podemos ver el modelo de ejemplo en la Figura 7.

Ahora, el modelo mostrado en Figura 7 se puede utilizar en cualquier otra situación particular con características similares. En consecuencia, el ingeniero tendrá la referencia de un experto (quien hizo el patrón) para no partir de cero en un proyecto específico.

Fowler plantea modelos conceptuales generales de las reglas del negocio de la organización para que puedan ser reutilizados. Aunque, estos modelos son independientes de la tecnología, constituyen un elemento importante para la lógica de información que manejará el futuro sistema informático. De esta forma, Fowler se convierte en uno de los primeros en aplicar el reúso en ámbitos alejados de la tecnología (con mayor nivel de abstracción). A diferencia de Fowler, en esta tesis no planteamos modelos generales del negocio, sino modelos de representación de activos concretos (requisito individual, patrón

## ESTADO DE LA CUESTIÓN

de requisitos, patrón de tipos de requisitos) cuya instanciación sirva directamente para su reúso en un proyecto.

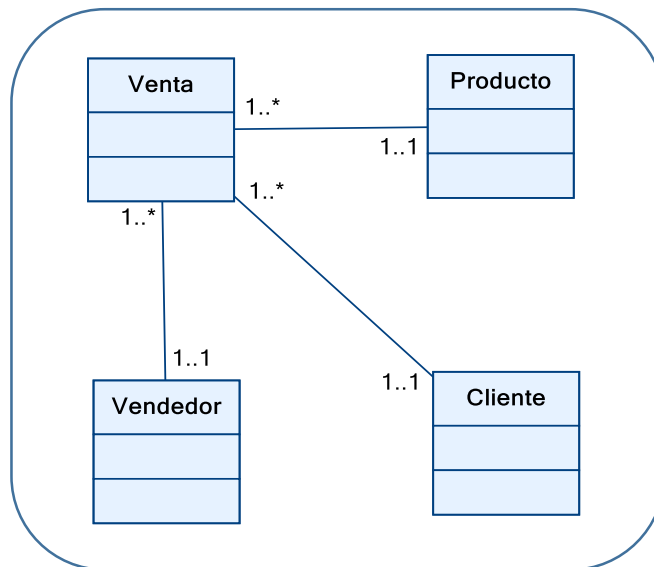


Figura 7 Ejemplo de modelo de un patrón de análisis según Fowler.

### 2.9.2 SIREN (Simple REuse of software requiremeNts)

El Grupo de Investigación en Ingeniería del Software del Departamento de Informática de la Universidad de Murcia (España) ha desarrollado el método SIREN (Toval, et al., 2002). Este método es un enfoque práctico para obtener y especificar los requisitos de un sistema software, basado en la reutilización de requisitos y en estándares de ingeniería del software.

El propósito del método SIREN es identificar descripciones del software que pueden ser reusadas (en su totalidad o en parte) con un mínimo de modificaciones, de manera que se reduzca el proceso de desarrollo de un proyecto. SIREN incluye un modelo de proceso en espiral, unas plantillas de documentos de requisitos por dominios y un repositorio de requisitos reutilizables.

Debido a que no es posible obtener un conjunto completo de requisitos en una sola iteración, SIREN considera que el proceso de ingeniería de requisitos se desarrolla a través de un modelo de ciclo de vida en espiral. En este modelo introduce algunas técnicas de reutilización en la fase de elicitación<sup>5</sup> y validación. Por ejemplo, tenemos el uso de plantillas para la especificación de requisitos, un repositorio para almacenar y recuperar requisitos, etc.

Los requisitos tienen principalmente formato textual. SIREN plantea una organización por dominios de los documentos de requisitos. Cada dominio se asocia a sus respectivas plantillas y patrones lingüísticos.

---

<sup>5</sup> Elicitar. Proviene del inglés *elicit*, hace referencia a la obtención de información de una fuente de manera provocada.

Para gestionar las características planteadas, SIREN propone un repositorio de requisitos reusables. La organización por dominios de documentos especificados según plantillas de requisitos, determinan la estructura del repositorio de requisitos reutilizables.

De igual forma se crea SirenTool (Nicolás, et al., 2003): herramienta de ingeniería de requisitos asistida por ordenador (CARE: Computer-Aided Requirements Engineering). SirenTool permite dar soporte automatizado a la reutilización de requisitos basado en el método de ingeniería de requisitos SIREN. En este sentido se analizó un conjunto de herramientas del mercado para obtener un conjunto de funcionalidades las cuales, junto con otras, se orientaron al soporte del método SIREN.

SIREN proponen un método general para la ingeniería de requisitos con reúso, plantillas específicas por dominios de requisitos y una herramienta automatizada de soporte. A diferencia de SIREN, en esta tesis planteamos un proceso específico del reúso de un activo de requisitos para que sea utilizada en la ingeniería de requisitos. En este proceso detallamos actividades de identificación, calidad para el almacenamiento, recuperación y adaptación en formato y contenido para un nuevo proyecto. Asimismo, planteamos que las plantillas de requisitos pueden ser configurables (no específicas por dominios) para la gran variedad de especificaciones que se pueden encontrar. Y aparte proponemos el reúso no solo de requisitos individuales, sino conjunto de requisitos que solucionen un problema mayor (patrones de requisitos) y también formas de organizar requisitos por tipos para una mejor gestión de los mismos (patrones de tipos de requisitos).

### **2.9.3 Reutilización de Requisitos en el modelo Mecano**

El grupo GIRO (Grupo de Investigación en Reutilización y Orientación a Objetos) de la Universidad de Valladolid propone el modelo Mecano para la reutilización de requisitos (Laguna, et al., 2001). Este modelo se basa en el almacenamiento en repositorios de Mecanos. Los Mecanos son descripciones de diferentes clasificaciones de requisitos individuales (llamados Assets) para tres niveles de abstracción (análisis, diseño e implementación). Los Mecanos, asimismo, extienden estas características de los Assets a las relaciones entre los mismos requisitos y otros elementos del modelo.

El grupo GIRO propone el reúso de Assets individuales o mediante los Mecanos específicos creados especialmente el reúso. Para su propósito, la reutilización de requisitos, GIRO se plantea el siguiente problema: encontrar un conjunto de requisitos reutilizables para un problema dado y la adaptación adecuada de los requisitos para futuros propósitos. Divide su solución respectivamente en:

- Formalización o normalización de los requisitos capturados mediante plantillas. Luego que los requisitos son almacenados en el repositorio de reúso. Estas plantillas facilitarán su tratamiento en cuanto a la relación con otros requisitos y otros elementos que abarcará un Mecano. De este modo se dispone de una forma directa de recuperar y combinar requisitos dentro del repositorio.
- Caracterización de los Mecanos. Es decir, la aplicación de descriptores funcionales del conjunto de requisitos reutilizables para algún propósito específico. Esta caracterización permitirá que se realice una mejor búsqueda de mecanos dentro del repositorio y por consiguiente la obtención de Assets respectivos.

- Parametrización de requisitos. A nivel de requisito individual se plantea la adaptación por parámetros, es decir, la adición de información adicional al requisito al momento del reuso.

También plantea un procedimiento no formalizado para el reuso de los requisitos. Parte de la captura de requisitos de usuario para compararlos con los del repositorio y encontrar los mecanos relevantes para el propósito buscado. Luego se entra en una fase de adaptación mediante la parametrización a múltiples situaciones de proyectos específicos.

El modelo Mecano propone el reuso de requisitos individuales (Assets) descritos con plantillas fijas. Asimismo, el modelo plantea el uso de descripciones de problemas que permiten la relación de Assets y otros elementos del modelo para obtener una mejor reutilización. En esta tesis doctoral planteamos el reuso de requisitos individuales con plantillas configurables debido a la distinta naturaleza de estos elementos. También a diferencia de la agrupación por Mecanos, planteamos Patrones de Requisitos, conjunto de requisitos finitos (no descripciones a las que se podrían incorporar requisitos) que solucionen un problema determinado del ámbito de la Ingeniería de Requisitos.

### **2.9.4 Patrones de Reutilización de Requisitos**

Investigadores del Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática y Estadística de la Universidad de Sevilla exponen resultados de la aplicación de patrones para la reutilización de requisitos ( (Durán, et al., 2000), (Durán, et al., 1999)).

El número de propuestas de reutilización en ingeniería de requisitos es aún escaso, sobre todo a nivel de requisitos de cliente o requisitos-C, normalmente expresados en lenguaje natural. En esta situación, la utilización en más de 80 prácticas académicas y en tres proyectos reales de las plantillas y patrones lingüísticos para requisitos-C, ha puesto de manifiesto la posibilidad de aplicar técnicas de reutilización durante la fase de ingeniería de requisitos.

Como resultado de esta experiencia se han identificado requisitos que, con pequeñas variaciones, aparecen en un elevado número de desarrollos. A dichos requisitos, una vez generalizados al eliminar los detalles específicos, los denominan patrones de reutilización de requisitos, o abreviadamente, patrones-R.

Entre otros, los resultados más interesantes de la normalización del formato de los requisitos, ha sido la posibilidad de compararlos e identificar patrones de reutilización, tanto a nivel de requisitos de cliente (requisitos-C, normalmente expresados en lenguaje natural) como a nivel de requisitos de desarrollador (requisitos-D, habitualmente modelos conceptuales), que facilitan el desarrollo y mejoran la calidad de las especificaciones de requisitos.

Las relaciones de trazabilidad realizadas entre requisitos-C, requisitos-D, e incluso elementos de menor nivel de abstracción como componentes software, ha permitido también plantear la posibilidad de reutilizar estructuras complejas, desde requisitos-C hasta código, obteniendo así una reutilización vertical que abarque distintos niveles de abstracción del desarrollo de software.

En un futuro esperan identificar nuevos patrones-R, sobre todo a partir de la disponibilidad del prototipo de una herramienta automatizada de ingeniería de requisitos que han desarrollado como parte del proyecto. Esta herramienta permitirá el tratamiento automatizado de las especificaciones de requisitos, así como el desarrollo de asistentes automatizados que ayuden a aplicar las ideas presentadas.

Durán plantea patrones de requisitos. Este patrón consiste en un solo requisito generalizado en una plantilla para que pueda ser reutilizado. En la presente tesis doctoral proponemos una diferente definición del Patrón de Requisitos: un conjunto de requisitos específicos que solucionan un problema determinado del ámbito de la ingeniería de requisitos. Y en cuanto a nuestro activo de requisito individual, planteamos que es un requisito específico potencialmente reutilizable, no una descripción generalizada.

### **2.9.5 Patrones de requisitos de software (Withall, 2010)**

Stephen Withall es licenciado en Ciencias Matemáticas en la universidad de Bristol, Reino unido. Tiene una amplia experiencia, se ha dedicado a la industria del Software por más de 26 años en diversos roles del ámbito del desarrollo de software y en diversas empresas de muchos países. Esta experiencia lo ha conducido a desarrollar muchas competencias y habilidades prácticas para el desarrollo de software (Withall, 2014).

Withall, entre otras cosas ha escrito un libro y un artículo orientado al reúso de requisitos en el desarrollo de software. Propone un conjunto de patrones de requisitos software que ayudan en gran medida a escribir determinados tipos de requisitos. Estos patrones permitirían a los analistas no gastar tanto tiempo en el desarrollo de la especificación de requisitos.

En la búsqueda de una solución para el tedioso trabajo de especificación, Withall propone en sus publicaciones un conjunto de 37 patrones de requisitos (Figura 8). Estos patrones son técnicas probadas para el reúso de requisitos. Los patrones consisten en modelos con plantillas para escribir mejor determinados tipos de requisitos. Además, adiciona a los patrones sugerencias de uso y ejemplos específicos.

En los patrones se diferencian determinadas secciones con funciones específicas. En la sección de Aplicación describe las situaciones favorables y situaciones desfavorables para el uso del patrón. En la sección de Discusión describe el proceso de uso del patrón y de obtención de información para el requisito. En la sección de Consideraciones para el diseño, brinda consejos para los diseñadores y programadores. Y en la sección de Contenido se especifican las demás características del requisito.

Se considera que un alto porcentaje de las características de cualquier sistema comercial son comunes a cualquier tipo de sistemas. Y solo una parte relativamente pequeña diferencia sistemas para fines muy diferentes. Withall considera que en la práctica el reúso de requisitos puede abarcar más de la mitad de los requisitos en una aplicación, pero es difícil calcular esto, pues también está el hecho de que un requisito de mayor complejidad puede influir mucho más en el tiempo.

# ESTADO DE LA CUESTIÓN

Withall propone plantillas generales que ayudan a escribir mejor determinados tipos de requisitos y adiciona sugerencias de uso. En la presente tesis doctoral planteamos requisitos específicos potencialmente reusables y además conjunto de requisitos (a los que llamamos patrones) que solucionan un problema específico de la Ingeniería de Requisitos para que sean directamente reusados en un nuevo proyecto.

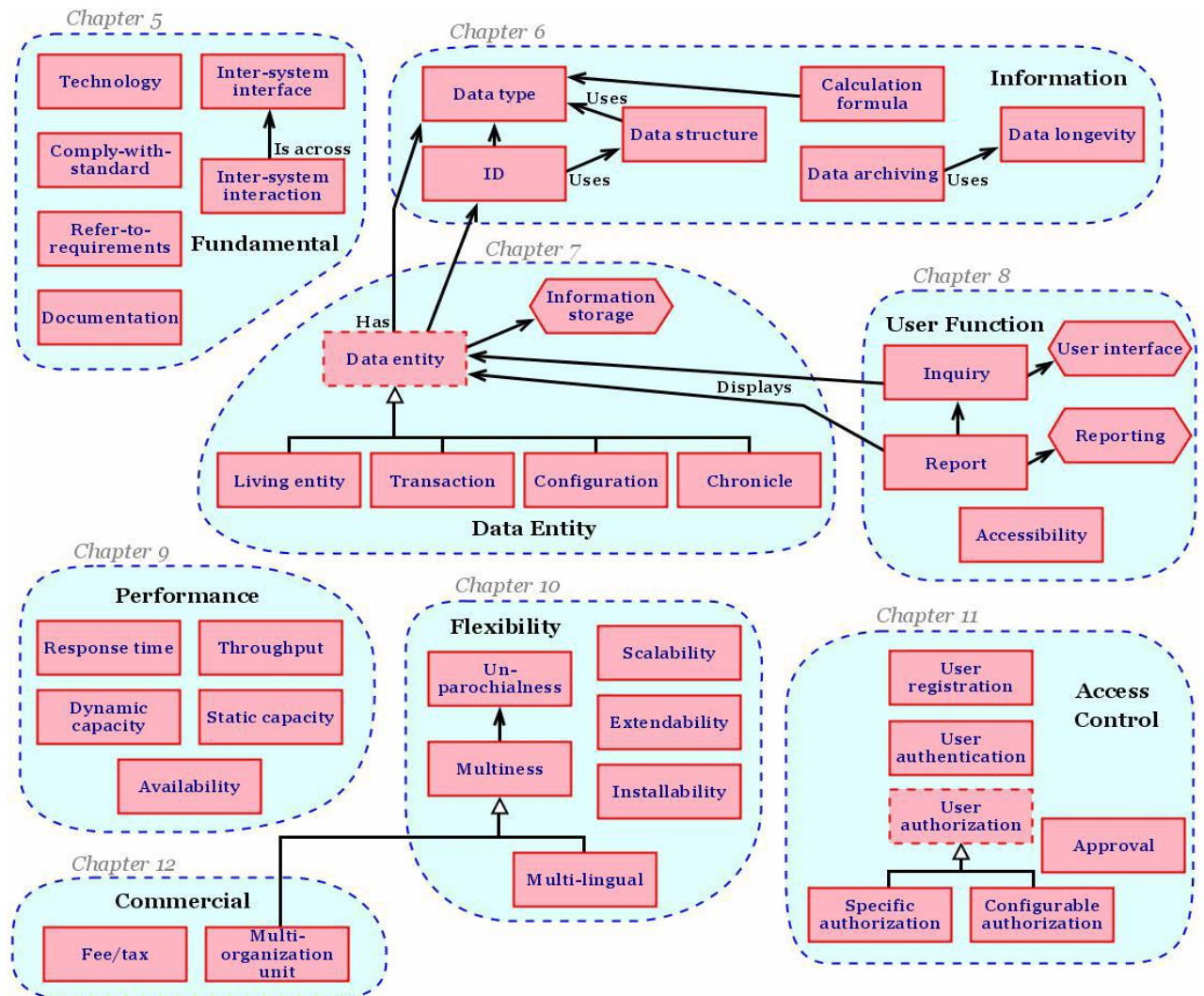


Figura 8 Patrones de requisitos software de Withall.

## 2.9.6 Técnicas de reúso: aplicación de dominios.

Johanna Suárez, de Unidades Tecnológicas de Santander, y Luz Gutiérrez de la Universidad Santo Tomás, de Colombia presentan el trabajo de investigación titulado: “*Tipificación de Dominios de Requerimientos para la Aplicación de Patrones Arquitectónicos*” (Suárez & Gutierrez, 2016). El presente trabajo propone un dominio de requerimientos que facilite la elección de patrones de arquitectura para proyectos específicos. Esto permitiría reducir la brecha de experiencia entre arquitectos seniors y arquitectos nóveles requeridos para el desarrollo de un proyecto.

La investigación consistió en seleccionar un grupo de proyectos de desarrollo de software para estudiar el conjunto de activos que generaban, sobretudo en cuanto a la fase

de requisitos y de arquitectura. De la revisión de los activos de requisitos se obtuvo el conjunto de requisitos funcionales y no funcionales de cada proyecto. Luego se seleccionaron los requisitos transversales a todos los proyectos estudiados. Con estos requisitos se definió un dominio de requerimientos según particularidades en común con varios sistemas. Por ejemplo, para los requisitos funcionales tenemos requisitos de autenticación, autorización, importación de datos, exportación de datos, asignación de funcionalidades, abstracción de sistemas de datos relacionados, entre otros. Mientras que los no funcionales pueden ser incluidos en seguridad, mantenibilidad, tiempo de respuesta, disponibilidad, entre otros. A partir del dominio se relacionaba un de sus requisitos (con similitudes a un proyecto), con un patrón de arquitectura idóneo para poder aplicarlos al nuevo proyecto.

Otro trabajo en la línea de dominios es el titulado “Reúso de Software Orientado a Dominios” (Burccella, et al., 2013), presentado por el Grupo “GIISCO Research Group” conformado por Agustina Buccella, Juan Luzuriaga, Alejandra Cechich, Rodolfo Martínez, Rafaela Mazalu, Marcis Cruz, Matias Pol’la, Maximikiano Arias y Adriana Martin, quienes pertenecen al “Departamento de Ciencias de la Computación” de la Universidad Nacional del Comahue – Neuquen, Argentina.

En este trabajo se plantea la propuesta de crear técnicas y herramientas (lineamientos o elementos reutilizables) comunes a un determinado dominio, para que puedan ser reutilizadas en proyectos relacionados el dominio respectivo.

El grupo primero identificó y acotó dos dominios: Sistemas de Información Geográficos y Aplicaciones para el Gobierno Electrónico. Luego crearon un conjunto de lineamientos y artefactos reutilizables para estos dominios específicos que permitieran el reúso efectivo de estos elementos en nuevos proyectos dentro del ámbito de los dominios correspondientes.

MORORE no clasifica los requisitos u otros elementos por dominios, sino propone, entre otras características, la representación de activos que puedan ser usados en diferentes proyectos, sin tener en cuenta el ámbito al que pertenezcan dichos proyectos.

### **2.9.7 Reúso efectivo de modelos domóticos a través de requisitos genéricos**

El grupo de investigación conformado por María Francisca Rosique Contreras, Diego Alonso Cáceres, Pedro Sánchez Palma, Francisco Sánchez Ledesma, del “Departamento de Tecnologías de la Información y las Comunicaciones - ETSI Telecomunicaciones” de la Universidad Politécnica de Cartagena (Murcia), presentan el trabajo denominado “Reúso efectivo de modelos domóticos a través de requisitos genéricos” (Rosique, et al., 2014).

Esta investigación presenta un enfoque que permite incrementar el reúso de modelos a través de requisitos genéricos. La investigación parte de un conjunto de requisitos genéricos del dominio de la domótica. Estos requisitos previamente catalogados se asocian individualmente a una especificación gráfica, representada en un Lenguaje de Específico de Dominio (DSL). Esta especificación gráfica representa parte de un modelo de domótica. De esta manera, al reusar los requisitos, indirectamente se reutilizan fragmentos del modelo domótico.



En el presente trabajo se reutilizan requisitos tipo del dominio de la domótica, como medio para reutilizar modelos específicos del ámbito domótico. En este sentido, MORORE se diferencia proponemos el reúso de activos específicos de cualquier ámbito, listos para ser reusados con mínimos cambios.

### **2.9.8 Evaluation of a systematic approach to requirements reuse**

Los investigadores Fabiane Barreto Vavassori Benitti y Rodrigo Cezario da Silva pertenecientes a la Universidade do Vale do Itajaí (Brasil) en su artículo “Evaluation of a Systematic Approach to Requirements Reuse” (Vavassori & Cezario, 2013). evalúan la problemática de la comunidad de Ingeniería de Software frente al tiempo utilizado en elaborar los requisitos de un proyecto.

Presentan un enfoque para la reutilización de los requerimientos, apoyado por una herramienta que genera sugerencias para la reutilización de los patrones de requisitos. El enfoque propuesto se basa en tres técnicas para propiciar sugerencias de reutilización; (I) Patrones de escritura de requisitos, (ii) catálogo de patrones y (iii) trazabilidad.

Se desarrolló un catálogo considerando treinta y siete patrones organizado en seis dominios distintos, como sigue: Base de datos lógica, interfaces externas, atributos del sistema, funcional, restricciones de diseño y rendimiento. Además, para facilitar el uso de patrones, el catálogo se organiza con las siguientes pautas; nombre, objetivo, contexto, problema, fuerzas, plantilla y ejemplos.

El proceso de reutilización se produce de una manera iterativa, es decir, los requisitos del sistema deben especificarse en múltiples ciclos, hasta que todos los requisitos del proyecto estén vinculados al final del proceso generando la especificación de requisitos de artefacto (documento de especificaciones). Las sugerencias de reutilización se producirán a partir del uso de un patrón.

Esta investigación propone patrones, como lineamientos que deben seguir los requisitos específicos de un proyecto. Asimismo, se enfoca en la trazabilidad que deben tener los requisitos entre sí que optimicen la especificación final. A partir de estos lineamientos, una herramienta automática realiza las sugerencias respectivas. MORORE también propone patrones, pero no como lineamientos que deba de seguir una especificación de requisitos. Los patrones de MORORE son un conjunto de requisitos específicos, contruidos a partir de un modelo, y almacenados para que puedan ser recuperados e integrados a un proyecto directamente con mínimos cambios.

### **2.9.9 Otros estudios:**

Después de la revisión bibliográfica, encontramos una multitud de estudios, empezando por McIlroy, que apuestan por la aplicación del reúso en el proceso de desarrollo de software. En la Tabla 12 presentamos estos otros estudios realizados referentes al tema general de la presente tesis doctoral: el reúso de requisitos.

## ESTADO DE LA CUESTIÓN

Autor	Descripción	Referencia	Tema
Barry Boehm	Demuestra que la estrategia de reutilización supone un ahorro de casi la mitad del esfuerzo invertido, alrededor del 47%.	(Boehm, 1993), (Boehm, 1999)	Reúso
Iam Sommerville	Resuelve que los beneficios de la reutilización son mayores cuando se incrementa el nivel de abstracción, es decir no sólo activos software a nivel de código sino activos software de diseño o especificación de requisitos.	(Sommerville, 2001)	Reúso
Gerald Kontoya	Los requisitos de restricción de un dominio determinado, los requisitos de presentación de datos y requisitos de políticas de organización son potencialmente reutilizables, y pueden constituir hasta el 50% de los requisitos de un nuevo proyecto.	(Kotonya, 1998)	Reúso de requisitos
Suzanne Robertson	La reutilización tiene ventajas cuantitativas: si se parte de requisitos que ya han sido especificados para otros proyectos se puede reducir el tiempo de producir dicha especificación y por tanto de todo el proceso de desarrollo de software.	(Robertson & Robertson, 2012)	Reúso de requisitos
Konrad, Cheng	Propone patrones de requisitos orientados a aplicaciones de sistemas embebidos de la industria.	(Konrad & Cheng, 2002)	Reúso de requisitos
Srivastava	Define plantillas de requisitos más un repositorio orientado al desarrollo de aplicaciones WEB.	(Srivastava, 2013)	Reúso de requisitos
Neighbors, Papamargaritis-Sutcliffe	Proponen modelos de dominios. Preparación previa de los activos software para su posterior reúso.	(Neighbors, 1984), (Sutcliffe, et al., 2006)	Reúso
Basili	Propone un proceso de recuperar: identificar, evaluar y seleccionar los activos software. Usan técnicas de búsqueda, recuperación y procesamiento de lenguaje natural.	(Basili & Musa, 1991)	Reúso
Cybulski, Reed	Mapeo de dominios. Usa PLN y conocimiento de dominios específicos.	(Cybulski & Reed, 2000)	Reúso
Mili	Clasificación manual de los descriptores en facetas.	(Mili, et al., 1997)	Reúso
Happel	Ontologías como contenedores estructurados del conocimiento para mejorar las búsquedas.	(Happel, et al., 2006)	Reúso
McIlroy	Uno de los primeros en señalar la relevancia de la reutilización como factor decisivo para mejorar la calidad del software y reducir los costes de desarrollo y mantenimiento.	(McIlroy, 1968)	Reúso

*Tabla 12 Otros estudios realizados respecto al reúso de requisitos.*

En resumen, las propuestas encontradas en la revisión de la bibliografía proponen plantillas, modelos generales de especificación o requisitos tipos que permiten el reúso de requisitos software. Al igual que estas propuestas, MORORE también propone el reúso aplicado al ámbito de la ingeniería de requisitos mediante un repositorio entre otras características. Pero las características diferenciales fundamentales que planteamos con MORORE las detallamos a continuación:

## ESTADO DE LA CUESTIÓN

---

- La representación de tres niveles de activos para el reúso del ámbito de los requisitos:
  - Requisito en forma individual: el modelo para un requisito específico, con plantilla configurable, parametrización para el reúso, elementos relacionados al requisito.
  - Patrones de requisitos: modelo para un conjunto de requisitos específicos relacionados y reusables en conjunto, plantilla descriptiva del conjunto, elementos relacionados al patrón.
  - Patrones de estructuras de tipos de requisitos: modelo para representar una estructura reusable de tipos para una mejor organización de requisitos, plantilla descriptiva de la estructura, elementos relacionados a la estructura.
- Un modelo para evaluar la calidad que consta de factores y métricas específicos que se aplican en el proceso de creación y uso de los productos reusables de cada uno de los niveles de MORORE. Este modelo de calidad permitirá reusar los activos propuestos con un mínimo de calidad.
- La definición de un proceso para el reúso de los activos propuestos que permita la indexación y recuperación de los productos reusables para integrarlos a un nuevo proyecto de desarrollo de software.

# **SOLUCIÓN PROPUESTA**

## 3 Solución propuesta

### 3.1 Aspectos generales

Ya mencionamos que los principales defectos de la ingeniería de software tienen su origen en una deficiente ingeniería de requisitos. En tal sentido hoy en día existen muchas investigaciones que aportan mejoras a esta disciplina. Estas investigaciones, vistas en el estado del arte, que pretenden la mejora de la ingeniería de requisitos las separamos en tres ámbitos:

- **Estándares.** El desarrollo de muchos estándares de requisitos reglamenta formas de especificación y procesos, con el objetivo de mejorar la ingeniería de requisitos. Proponen numerosas plantillas que facilitan la especificación de los requisitos, además sugieren los pasos a seguir para llevar el proceso de la ingeniería de requisitos con éxito.
- **Herramientas.** Existen también muchas herramientas ideadas para dar soporte a los procesos y activos relacionados con la ingeniería de requisitos. Estas herramientas traen propuestas muy interesantes para mejorar la ingeniería de requisitos
- **Técnicas y métodos.** Tenemos también muchas técnicas y métodos que pretenden aportar mejoras a la disciplina de la ingeniería de requisitos. Así, por ejemplo:
  - Intentos de definir la ingeniería de requisitos y por tanto definir su ámbito de acción dentro de todo el proceso de desarrollo de software.
  - Diferentes propuestas metodológicas que describen un conjunto de fases y actividades para conducir de manera aceptable el proceso de ingeniería de requisitos.
  - Aplicación de técnicas de reúso en el ámbito de la ingeniería de requisitos.

Las técnicas de reúso surgen por la necesidad de la industria de producir mejores productos y economizar el proceso de desarrollo. En el ámbito de la ingeniería de software se define el reúso como la utilización de cualquier tipo de activo, o parte del mismo, creado con anterioridad, en un nuevo proyecto (IEEE, 1990).

Respecto a la informática en general, en sus inicios el reúso se aplicó con excelentes resultados en la industria del hardware. La fabricación de hardware era un proceso en muchos casos automático que consiste en el ensamblado de partes estándar. Pero al empezar hablar del reúso en el software surgen los problemas.

El reúso del software en sus inicios se limitaba la mayoría de las veces al reúso de código, y no se tenían en cuenta los muchos subproductos que surgen durante todo el proceso de desarrollo de software. El código por sí mismo tenía problemas cuando se deseaba aplicar en diferentes plataformas y también se volvía más problemático por la aparición de nuevas tecnologías (TRC, 2017). No obstante, el reúso de código fue mejorando a medida que se elevaba el nivel de abstracción a objetos software y operaciones (componentes), sin preocuparse de la implementación del código a bajo nivel.

Esta evolución del reúso a nivel de componentes empezó a ser aceptada y utilizada, los ingenieros desarrollaban componentes, los almacenaban y reusaban. Esto trajo consigo otro problema: una vez que se tenía un repositorio relativamente grande, era difícil encontrar el componente adecuado a una situación específica (Baeza-Yates & Ribeiro-Neto, 1999). Esta limitación en la búsqueda y por consiguiente recuperación del componente adecuado motivó que se eclipsaran por un tiempo las técnicas de reúso (década de los noventa).

Las técnicas de reúso han retomado su presencia en el proceso de desarrollo de software, gracias al desarrollo de las técnicas de indexación y recuperación. Por ejemplo: análisis de dominios, facetas, procesamiento del lenguaje natural ( (Baeza-Yates & Ribeiro-Neto, 1999), (Prieto-Díaz, 1987), (Manning, et al., 2008), (Büttcher, et al., 2010), (Campos, et al., 2015), (Rini & Govilkar, 2016), (Kancheva & Anand, 2016)), etc. Estas técnicas ayudan a superar el problema de recuperación de los activos software.

Las técnicas de reúso toman mayor importancia en la actualidad pues se están aplicando a activos de mayor nivel de abstracción, por ejemplo del ámbito: del diseño de software (Gamma, et al., 1995) de la arquitectura de software (Buschmann, et al., 2007) e incluso en el ámbito de la ingeniería de requisitos ( (Withall, 2010), (Hurtado, et al., 2005), (Durán, et al., 2000), (Fowler, 1997), entre otros). Sommerville menciona que reusar activos con un nivel de abstracción mayor, en el proceso de desarrollo de software, tiene mucho más valor y beneficios (Sommerville, 2001). Mientras estas técnicas se apliquen en activos software más independientes de la plataforma, evitaremos los problemas asociados a ésta: diversidad, obsolescencia, etc. (TRC, 2017).

En consecuencia, la presente tesis doctoral se basa en la línea de investigación referente a la aplicación de técnicas de reúso al ámbito de la ingeniería de requisitos. Específicamente desarrollamos un modelo de requisitos orientado al reúso efectivo denominado MORORE. Este modelo aplica técnicas de reúso y se orienta a uno de los ámbitos de mayor nivel de abstracción del proceso de desarrollo de software, la ingeniería de requisitos. En tal sentido, con este modelo pretendemos brindar una contribución para el mejoramiento sustancial de la ingeniería de requisitos y por consiguiente de todo el proceso de desarrollo de software.

### **3.2 Modelo de requisitos orientado al reúso efectivo (MORORE)**

Según se ha descrito previamente, el objetivo de la presente tesis doctoral es el desarrollo de un modelo que permita el reúso de requisitos. Para tal propósito, MORORE se estructura en las siguientes partes, también mostradas en la Figura 9:

- La primera parte se orienta a la definición de tres niveles de activos software. Estos activos reusables relacionados con la ingeniería de requisitos son: requisitos individuales, patrones de requisitos y patrones de estructuras de tipos de requisitos.
- Asimismo, en una segunda sección, definimos un conjunto de factores y métricas que permiten evaluar la calidad de los activos software para su almacenamiento. Esta evaluación permite realizar un control de calidad de los activos software de MORORE al momento de ser almacenados para su posterior reúso, garantizando un mínimo de calidad en los elementos reusables.

# SOLUCIÓN PROPUESTA

- El otro punto adicional de MORORE es un proceso general que permite el reúso del activo software definido. Este proceso consta a su vez de dos partes: indexación y recuperación de los activos de MORORE.

Hay que resaltar que, los diagramas de las dos primeras secciones del modelo fueron revisados en un proyecto de fin de carrera (PFC) propuesto y dirigido por el autor de la presente tesis doctoral (Sánchez, 2016).

Como apreciamos en la Figura 9, en primera instancia tenemos los tres niveles de activos software reusables, en el centro el nivel de patrones de requisitos, al lado izquierdo los requisitos individuales y al lado derecho el patrón de estructuras de tipos de requisitos. En segunda instancia, las flechas indican que a cada uno de estos niveles le corresponde un conjunto de factores y métricas de calidad definidos específicamente. Y en la parte inferior, representamos el proceso general que podrá ser aplicado a cualquier activo software independientemente del nivel al que pertenezcan respecto al modelo.

Estas tres partes conforman el modelo de requisitos orientado al reúso de MORORE. Para utilizar MORORE, podemos aplicar estas partes en conjunto o en forma parcial. Por ejemplo, podemos utilizar solo el nivel de patrón de requisitos en un proyecto específico. El patrón pasaría por un control de calidad dentro del proceso definido para su reúso respectivo. En este caso no utilizaríamos el nivel de los patrones de estructuras de tipos ni todas las características del nivel de requisitos en forma individual.

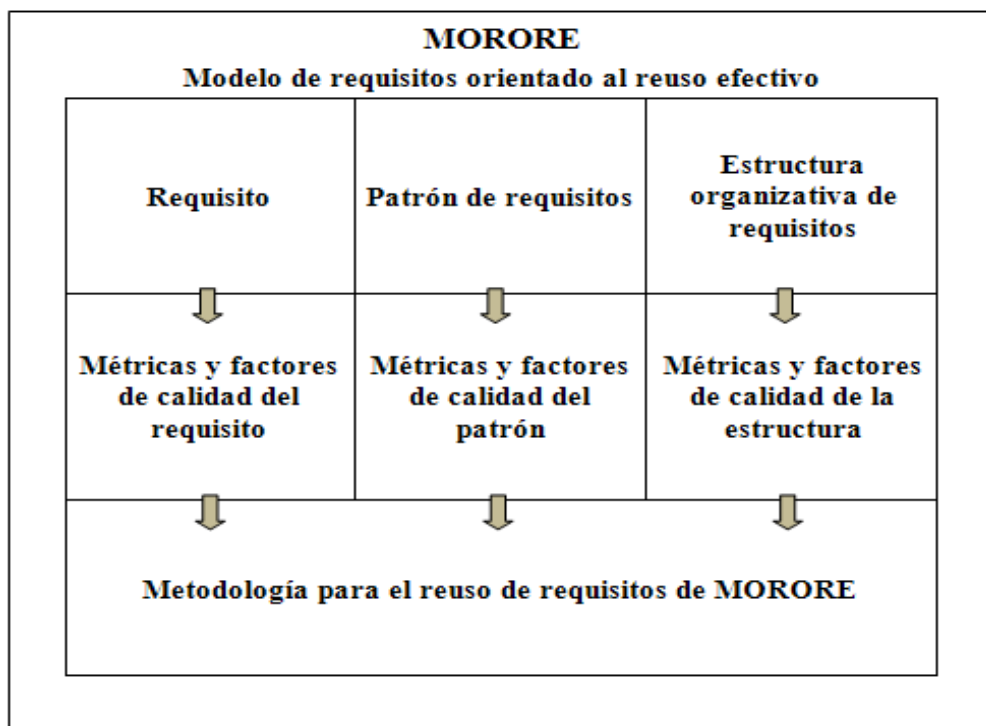


Figura 9 Esquema general del modelo de requisitos orientado al reúso efectivo (MORORE)

Con el **nivel de requisito individual** del modelo, planteamos la especificación particularizada de un requisito como activo reusable. Este nivel está especialmente pensado para tratar de manera individual una funcionalidad específica común de los sistemas de información. Esta orientación principal no impide que el modelo de requisito individual se utilice para reusar otros tipos de requisitos, como los no funcionales. Por

## SOLUCIÓN PROPUESTA

ejemplo planteamos en la Tabla 13 un requisito funcional potencialmente reutilizable y en la Tabla 14 un requisito no funcional.

Atributo	Valor
Identificador	RF-001
Descripción	El sistema debe permitir al usuario vendedor registrar los datos del cliente. Los datos del cliente para registrar deben ser los siguientes: nombre, apellidos, DNI, dirección, correo electrónico y teléfono
Tipo	Funcional
Autor	Omar Hurtado
Complejidad	Baja

*Tabla 13 Ejemplo de requisito funcional potencialmente reusable*

Tomando el ejemplo de requisito expuesto en la Tabla 13, apreciamos un conjunto de atributos cuyos valores describen las características del requisito. Más adelante describiremos las demás características de MORORE para este nivel (relaciones con otros elementos, parametrización, etc.). Por lo pronto sólo pretendemos, con este ejemplo reducido, aclarar a qué tipo de activo software orientamos nuestro estudio en este nivel.

Atributo	Valor
Identificador	RNF-0015
Descripción	El sistema debe permitir registrar los valores monetarios con dos decimales.
Tipo	No funcional
Autor	Omar Hurtado
Complejidad	Baja

*Tabla 14 Ejemplo de requisito no funcional potencialmente reusable*

El **nivel de patrón de requisitos** surge con la idea de Alexander (Alexander, et al., 1977), la cual tiene su origen en el ámbito de la arquitectura civil. Esta idea es posteriormente introducida en el ámbito de la informática, donde destaca principalmente Gamma (Gamma, et al., 1995) específicamente en los famosos patrones de diseño. Nosotros aprovechamos esta idea para definir los patrones de requisitos de MORORE.

Definimos nuestro patrón como un conjunto de requisitos relacionados entre sí, junto con otros elementos asociados, que han sido probados con éxito como solución a un problema recurrente dentro del ámbito de la ingeniería de requisitos. Por ejemplo planteamos en la Tabla 15, la descripción de un patrón según MORORE. Sólo planteamos en el ejemplo un patrón con algunos atributos y sus valores, y mencionamos algunos requisitos constitutivos (en este caso, tres requisitos). Posteriormente detallamos todas las características del patrón de MORORE (total de sus atributos, relaciones con otros elementos, etc.).

También aplicamos la idea de los patrones al **nivel de estructuras de tipos** de requisitos de MORORE. Muchos estándares proponen diferentes estructuras de tipos. Las mismas organizaciones de desarrollo de software tienen sus propias formas para organizar requisitos dependiendo del tipo de proyecto u otras características. Nuestro modelo pretende almacenar y reusar estas estructuras de tipo según sea la situación más adecuada para un proyecto específico.



## SOLUCIÓN PROPUESTA

Atributo	Valor
Identificador	PS-001
Clasificación	Seguridad
Nombre	Nivel 2 de acceso a un sistema
Propósito	Especifica los requisitos para definir la seguridad en el acceso a un sistema de información. Este grado de acceso debe considerar las restricciones del nivel 2 según la clasificación de dificultad de la empresa Asyncronyx <sup>6</sup> .
Aplicabilidad	Aplicable a software de gestión comercial de nivel operacional (cajeros de tiendas comerciales, farmacias, almacenes, etc.)
Elementos	
Requisito 01	El sistema debe tener dos niveles de seguridad para el acceso: Nombre de usuario y clave.
Requisito 02	El nombre de usuario debe ser el identificador del empleado dentro de la organización.
Requisito 03	La clave de acceso debe tener como mínimo 8 caracteres. Entre estos caracteres debe haber como mínimo una letra y un dígito.
Fichero relacionado	Norma de nivel 2 de la empresa Asyncronyx.

*Tabla 15 Ejemplo de patrón de requisitos*

Como ejemplo planteamos en la Tabla 16, la descripción de una estructura de tipos con atributos y valores elementales. Luego, en el apartado correspondiente, describiremos detalladamente todas las características de la estructura de tipos de MORORE.

Atributo	Valor
Identificador	EO-001
Nombre	Estructura de tipos de Métrica V3
Descripción	Describe la organización que plantea Métrica V3. Consta de un sólo nivel jerárquico, y cada uno de los requisitos sólo puede pertenecer a un nivel.
Elementos	
Requisitos funcionales (1)	
Requisitos de rendimiento (2)	
Requisitos de seguridad (3)	
Requisitos de implantación (4)	
Requisitos de disponibilidad de sistema (5)	
Enlace relacionado	<a href="http://www.csae.map.es/csi/metrica3/index.html">http://www.csae.map.es/csi/metrica3/index.html</a>

*Tabla 16 Ejemplo de estructura de tipos*

El otro aspecto planteado para lograr un reúso de requisitos es, la definición de factores y métricas de calidad para cada nivel de activos definido. Estos factores y métricas de calidad permiten que nuestro repositorio contenga activos software con un mínimo de calidad. En consecuencia, al reusar activos software de probada calidad incrementamos los beneficios en cuanto a tiempo, recursos, calidad, etc., para futuros proyectos. En MORORE definimos un conjunto de factores de calidad con sus respectivas métricas independientemente para cada nivel de activos software.

<sup>6</sup> Información ficticia, es sólo creada para dar a entender el ejemplo del patrón.

## SOLUCIÓN PROPUESTA

En la Tabla 17 planteamos algunos ejemplos de factores y métricas de calidad para los activos software de los diferentes niveles de MORORE. Detallamos y explicamos todos los factores, métricas, otros elementos y sus respectivas relaciones, definidos para medir la calidad de todos los niveles de activos de MORORE en secciones posteriores.

Finalmente, en la tercera instancia del modelo, planteamos un proceso que describa los pasos que deberán seguir los ingenieros para lograr el reúso de los activos. Este proceso consta de dos fases: indexación y recuperación. Cada una de estas fases se subdivide en pasos los cuales describiremos más adelante. Este proceso es común para cualquier activo software independientemente del nivel del modelo. En consecuencia, aunque cada activo software del modelo tiene sus propias características y por tanto su propia manera de especificación, existe la posibilidad de recuperar, ante una misma consulta, diferentes activos software de diferentes niveles. Esta particularidad de MORORE permite optimizar la búsqueda de los activos software requeridos para una situación específica.

Nivel	Factor	Métricas
Requisito individual	Comprensibilidad	Cantidad de términos imprecisos, cantidad de signos de puntuación, existencia de términos conectivos, etc.
	Compleitud	Tamaño de la frase en palabras, cantidad de dependencias con otros elementos, cantidad de solapamientos entre requisitos, cantidad de elementos condicionales, etc.
Patrón de requisitos	Aplicabilidad	Cantidad de usos exitosos, cantidad de ámbitos de uso, cantidad de elementos parametrizables, etc.
	Comprensibilidad	Cantidad de términos impresos en la descripción problema-solución, cantidad de signos de puntuación en la descripción del problema-solución, etc.
Estructura de tipos	Complejidad	Cantidad de niveles jerárquicos, Cantidad de relaciones no jerárquicas entre los subniveles, etc.
	Compleitud	Cantidad de aspectos cubiertos, cantidad de dependencias entre los elementos, cantidad de solapamientos, etc.

*Tabla 17 Ejemplo de factores y métricas de calidad de los niveles de MORORE*

### 3.2.1 Nivel de requisito individual

Para realizar la descripción específica de MORORE planteamos en primer lugar el nivel de requisito individual. Este nivel es uno de los tres tipos de activos software definidos en MORORE. El nivel individual de requisito es concebido para soportar la especificación correcta y completa de un requisito.

En la Figura 10, mediante un diagrama de clases de UML (Unified Modeling Language – Lenguaje unificado de modelado<sup>7</sup>), especificamos los elementos que componen la representación del nivel de requisito individual para MORORE.

<sup>7</sup> UML. lenguaje gráfico y formal para el modelado de sistemas software. Es un estándar respaldado por la OMG (Object Management Group), el cual es el más conocido y utilizado en la actualidad (UML, 2017).

# SOLUCIÓN PROPUESTA

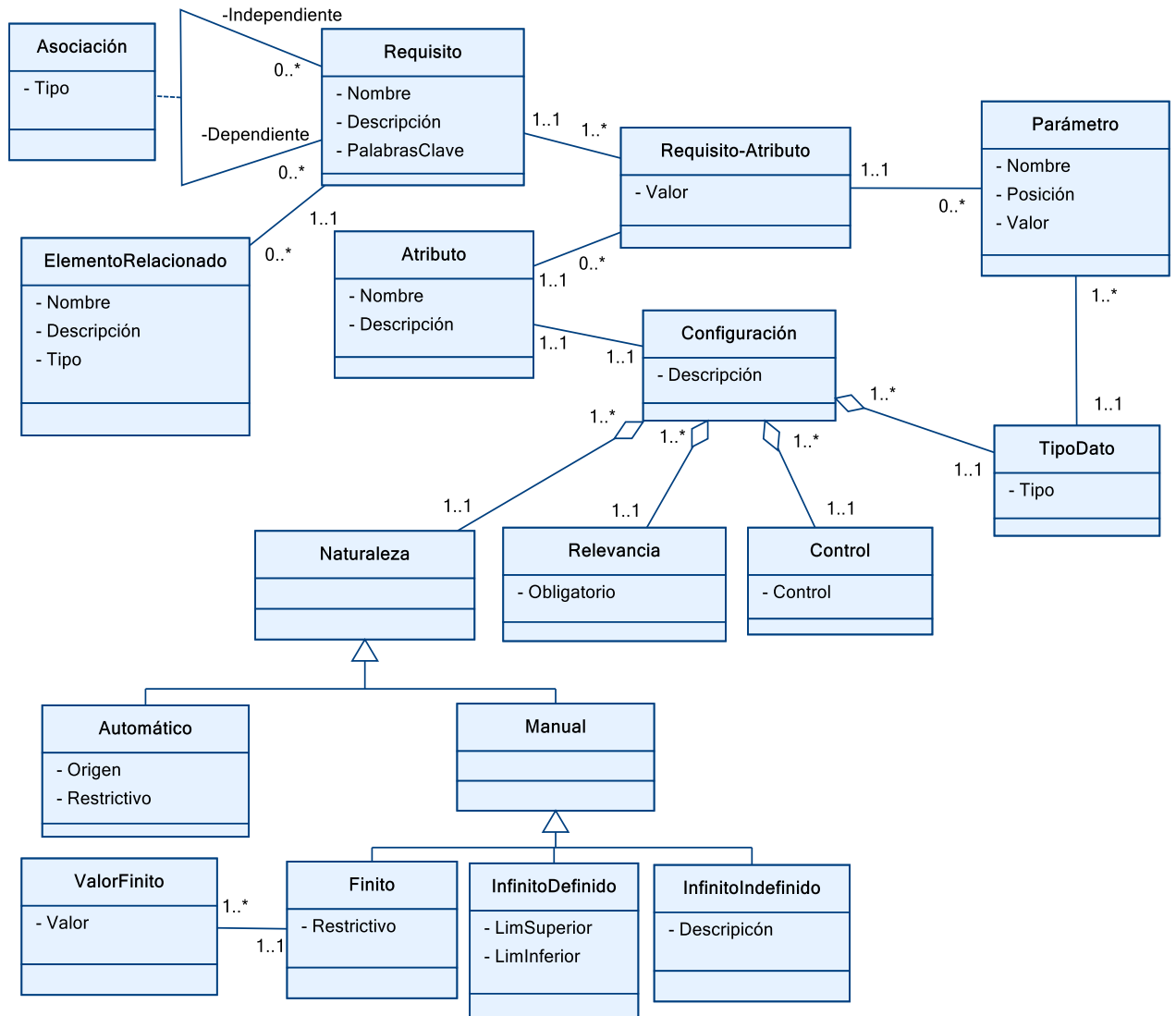


Figura 10 Modelo del nivel de requisito individual de MORORE

Aclaremos que esta representación soporta la descripción de cualquier tipo de requisito de especificación en lenguaje natural. Permite la representación de las características de la funcionalidad de un software, así como de las restricciones del mismo.

El nivel de patrón representa un conjunto de requisitos interrelacionados con el propósito de brindar una solución a un problema recurrente, mientras que el nivel de patrón de estructuras de tipos proporciona diversas formas de ordenar a los requisitos. En consecuencia, el nivel de representación de requisito individual puede estar contenido en los niveles más complejos de MORORE. Respecto a esta jerarquía de los elementos del modelo, el requisito individual se convierte en el nivel más básico de activo reusable de todo el modelo.

Otro punto importante del modelo, respecto al nivel de requisito individual, es su aplicación en una herramienta CASE, específicamente las herramientas tipo CARE (Computer-Aided Requirements Engineering – Ingeniería de Requisitos Asistido por

Computador<sup>8</sup>). Aunque el alcance de la presente investigación es el desarrollo del modelo de reúso de requisitos, consideramos aplicar en el futuro, las ideas del modelo en una herramienta CARE. En este sentido, la descripción del modelo está orientada a la aplicación del mismo en una herramienta automática.

Para una mejor comprensión del modelo, la descripción la dividimos por áreas funcionales. Estas áreas representan las características generales de este nivel orientada a su implementación en una herramienta automática. Estas áreas funcionales son:

### **3.2.1.1 Plantilla de requisitos configurable**

Hay que recalcar que, esta sección, está basada en un trabajo de investigación (Hurtado, et al., 2009) desarrollado por el autor de la presente tesis doctoral.

Es bien conocido que un producto Software tiene muchas diferencias respecto a otros productos de la industria. Coincidimos con Llorens (Llorens, 1996) cuando asigna la propiedad de ser dúctil al software. Entre otras cosas, esta propiedad hace que los proyectos del desarrollo de software tengan entre sí muchas diferencias en varios ámbitos.

En forma específica, las distintas necesidades de formas de especificación de requisitos se dan por varios motivos, así tenemos:

- Las distintas naturalezas y envergaduras de los proyectos de desarrollo de software (de donde podemos obtener requisitos reusables) requieren diferentes características o necesidades de detalle de la especificación de los requisitos.
- Los distintos tipos de requisitos (funcionales, no funcionales, del negocio, del proyecto, etc.) necesitan diferentes formas de especificación.
- Los distintos equipos de desarrollo de software tienen particulares formas de especificar los requisitos.
- Los estándares de especificación de requisitos proponen distintos atributos de especificación de requisitos.
- También las herramientas de gestión de requisitos tienen distintos atributos para la especificación de requisitos.

Después del estudio de estándares relacionados con la especificación de requisitos y el estudio de herramientas de gestión de requisitos, comprobamos que las diferencias también abarcan las formas de especificar los requisitos. Por consiguiente, para diferentes proyectos se tendrán diferentes necesidades de especificación de requisitos. En consecuencia, dependiendo de las circunstancias, se requerirán diferentes formas (plantillas) de especificación de requisitos. Estas diferentes formas de especificar requisitos se evidencian al momento de identificar o crear los requisitos para su posterior reúso.

La definición de MORORE, respecto al ámbito del requisito individual y orientado a una herramienta CASE, estará compuesta por una plantilla de especificación

---

<sup>8</sup> CARE es un tipo de herramienta CASE que se orienta de forma específica a brindar soporte automatizado al proceso de la ingeniería de requisitos.

## SOLUCIÓN PROPUESTA

configurable por el usuario mientras usa la herramienta. Durante el desarrollo de un requisito individual reusable, un ingeniero de software podrá configurar la plantilla de especificación del requisito. Este ingeniero de software será el usuario responsable de seleccionar los atributos de la plantilla que describirá al futuro requisito reusable. Esta función de plantilla configurable también es de mucha utilidad para la gestión de un proyecto de Software propiamente dicho.

A continuación, describimos primero los elementos centrales del modelo que forman parte de la plantilla de especificación configurable del nivel de requisito individual:

La clase **Requisito**. Es el elemento que representa en forma general a la plantilla configurable de especificación de requisitos. Por medio de esta clase, representamos el conocimiento necesario del requisito individual instanciado, que se almacena para su posterior reúso en un proyecto específico. A continuación, describimos los atributos que brindan la información general y rápida para el conocimiento y uso del requisito:

- **Nombre.** Representa al nombre corto del requisito para un mejor entendimiento del mismo al momento de su recuperación.
- **Descripción.** Este atributo se usa para especificar de manera objetiva qué hace y qué no hace el requisito y algunas características para su reúso. El contenido puede ser un resumen general informativo del requisito instanciado. Sirve para que el usuario tenga un conocimiento rápido del requisito para su posible reúso. Como los requisitos son creados para que otras personas los puedan utilizar después, es importante una descripción clara y fácil de entender, se deben evitar términos técnicos y ambiguos. No confundir con un posible atributo “Descripción”, donde se especifica detalladamente al requisito instanciado. Puesto que, el atributo “Descripción”, describe detalladamente a un requisito instanciado, la diferencia con el atributo **Descripción** de la clase **Requisito**, es principalmente el nivel de detalle entre ambos y alguna característica de cómo reusar el requisito en este último.
- **PalabrasClave.** Contiene un conjunto de palabras independientes que facilitan la búsqueda del requisito. Se recomienda que se utilicen entre tres y cinco frases separadas por comas.

*Tabla 18 Instancia de la clase Requisito*

Clase	Atributo	Valor de ejemplo	Descripción
Requisito	Nombre	“Registro de socios”	Nombre corto para una rápida ubicación.
	Descripción	“Permite el registro de datos de personas, para el caso, los socios de un club. Asimismo, se puede usar cuando se tenga que registrar los datos personales de una lista de personas.”	Descripción general para facilitar el conocimiento del requisito e información para su reúso.
	Palabras claves	“Socio, Registrar datos, Club, lista de personas.”	Palabras para facilitar la indexación y búsqueda del requisito.

En la Tabla 18, mostramos, como ejemplo, a una instancia de la clase **Requisito**.

## SOLUCIÓN PROPUESTA

La clase **Atributo**. Representa a cada atributo descriptivo de la plantilla que puede tener el requisito. Este atributo constituye una característica específica de requisito reusable. En tal sentido, para MORORE, el atributo de una plantilla es la propiedad descriptiva relevante para la especificación efectiva de un requisito.

En la Figura 10 apreciamos que la clase **Atributo** se relaciona con la clase “Requisito” por medio de la clase intermedia **Requisito-Atributo**, que describimos a continuación. Como ya especificamos, el conjunto de estas propiedades (atributos) descriptivas relevantes puede variar por el tipo de proyecto, por el tipo de requisito, por el equipo desarrollador, etc.

Los atributos que conforman la clase **Atributo** los especificamos a continuación:

- **Nombre.** Representa al nombre del atributo del requisito instanciado. Este elemento debe contener un nombre mnemónico, que sea fácil de recordar y que especifique en forma clara y concreta al tipo de atributo. Se recomienda que el nombre no exceda las cinco palabras.
- **Descripción.** Contiene la información necesaria para entender o conocer al atributo del requisito instanciado.

En la Tabla 19, mostramos, a modo de ejemplo, una instancia obtenida de la clase **Atributo**.

Clase	Atributo	Valor de ejemplo	Descripción
Atributo	Nombre	“Fecha de la última revisión”	Nombre del atributo de la plantilla.
	Descripción	“Este atributo contendrá una fecha. Es usado para indicar la fecha en que se realizó la última revisión del requisito en un proyecto”	Descripción general del atributo de la plantilla.

*Tabla 19 Instancia de la clase Atributo*

Después de la revisión de bibliografía, estándares y herramientas de gestión de requisitos, obtuvimos un conjunto suficiente de atributos potencialmente usables para la descripción del requisito instanciado. Este listado de atributos están especificados en la Tabla 20.

## SOLUCIÓN PROPUESTA

Atributo	Descripción
Identificador	Identifica unívocamente al requisito.
Nombre	Nombre corto del requisito.
Descripción	Descripción detallada del requisito
Autor	Nombre del creador del requisito.
Fuente	Nombre del origen de información para el requisito.
Stakeholders	Nombre de los interesados en el desarrollo del requisito.
Tipo	Clasificación del requisito.
Fecha de creación	Fecha en la que se creó el requisito.
Fecha de última revisión	Fecha en la que se hizo la última revisión del requisito.
Estado	Situación temporal en la que se encuentra el requisito.
Necesidad	Importancia del desarrollo del requisito.
Prioridad	Orden temporal de desarrollo del requisito.
Estabilidad	Probabilidad de cambio que pueda sufrir el requisito.
Complejidad	Dificultad para el desarrollo del requisito.
Costo	Costo para el desarrollo del requisito.
Versión	Orden de modificaciones al requisito.
Precondiciones	Situación o condición del sistema antes de ejecutarse la función definida por el requisito.
Post condiciones / resultado / objetivo	Situación o condición del sistema después de ejecutarse la función definida por el requisito.
Activador	Evento que inicia la funcionalidad definida por el requisito.
Excepciones	Situaciones excepcionales relacionadas con la función definida por el requisito.
Restricción	Limitación de la funcionalidad definida por el requisito (Cuando la restricción sólo afecta al requisito).

*Tabla 20 Atributos propuestos para la plantilla configurable de especificación de requisitos*

La clase **Requisito-Atributo**. Es la clase intermedia entre la clase **Requisito** y la clase **Atributo**. Esta clase representa al valor que toma cada tipo de atributo de un requisito instanciado. Esta clase consta del atributo siguiente:

- **Valor.** Representa al valor que se instanciará para un atributo que pertenezca a un requisito.

En la Tabla 21, mostramos, como ejemplo, a una instancia de la clase intermedia **Requisito-Atributo**.

Clase	Atributo	Valor de ejemplo	Descripción
Requisito-Atributo	Valor	"11/11/2015"	Valor de fecha específico que tomará el atributo del requisito.

*Tabla 21 Instancia de la clase intermedia Requisito-Atributo*

Los elementos que definimos a continuación, representan las propiedades necesarias para cada requisito, requeridas por el sistema informático, para la generación de la plantilla en forma automática.

La clase **Configuración**. Representa al conjunto de propiedades que se le asigna a un atributo de requisito para facilitar su creación en la plantilla automática. El conjunto de estas propiedades, asignadas a cada atributo de un requisito, permitirá la generación de la plantilla por el usuario mientras usa la herramienta. Esta clase consta del siguiente atributo:

## SOLUCIÓN PROPUESTA

- **Descripción.** Permite describir las características generales de la configuración formada por el conjunto de propiedades que se asigna a un atributo de un requisito.

En la Tabla 22, mostramos, como ejemplo, a una instancia de la clase **Configuración**, asumido para el atributo “Fecha de última revisión”.

Clase	Atributo	Valor de ejemplo	Descripción
Configuración	Descripción	“Configuración para un atributo tipo fecha, especificado por el usuario y no obligatorio”	Descripción general de la configuración.

*Tabla 22 Instancia de la clase Configuración*

A continuación, especificamos las clases que representan a las propiedades que forman parte de la clase **Configuración**. Cada propiedad permite establecer una característica orientada a la generación automática del atributo en la plantilla del requisito.

La clase **Naturaleza**. Define la forma de obtención del valor del atributo. Representa si el valor del atributo será automático o manual. Estas dos alternativas están representadas en dos subclases.

La subclase **Automático**. Permite especificar que el origen del valor puede ser tomado automáticamente. Por ejemplo: la fecha del sistema, un código secuencial, etc. Esta clase consta de los siguientes atributos:

- **Origen.** Este atributo representa la definición de la alternativa de posibles fuentes de origen de los datos automáticos.
- **Restictivo.** Atributo booleano que especifica: si es “restrictivo”, indica que el valor asignado no puede ser modificado. En cambio, si el atributo se señala como “no restrictivo”, aparecerá de forma automática y puede ser modificado por el usuario.

En la Tabla 23, mostramos, como ejemplo, a una instancia de la subclase **Automático**, asumido para la configuración de un atributo “Fecha actual”.

Clase	Atributo	Valor de ejemplo	Descripción
Naturaleza / Automático	Origen	“Fecha tomada del sistema”	Fecha tomada del sistema del ordenador.
	Restictivo	“No restrictivo”	Puede ser modificado.

*Tabla 23 Instancia de la subclase Automático*

La subclase **Manual**. Permite especificar que el origen del valor debe ser especificado por el usuario. Este origen a su vez puede ser de tres tipos:

- **Finito.** Representado como subclase de la clase **Manual**. Representa a un valor de un conjunto de valores alternativos específicos. Esta subclase consta de los siguientes atributos:
  - **Restictivo.** Atributo booleano que indica si el usuario podrá o no adicionar un valor. Por ejemplo: si es “sí”, entonces los valores posibles son fijos, el usuario no puede añadir ni eliminar ninguno. Si el valor es “no”, el usuario puede editar los valores alternativos.



## SOLUCIÓN PROPUESTA

- La subclase **Finito** está asociada a una clase **ValorFinito**. La clase **ValorFinito** representa la definición de las alternativas posibles de valores discretos para el atributo seleccionado.

En la Tabla 24, mostramos, como ejemplo, a una instancia de la subclase **Finito** y de la clase **ValorFinito**, por estar muy relacionadas, asumido para la configuración de un atributo “Estado”.

Clase	Atributo	Valor de ejemplo	Descripción
Naturaleza / Manual / Finito	Restictivo	“No”	El usuario puede modificar el valor (cambiar la fecha por ejemplo).
ValorFinito	Valor	“Propuesto”	Para un atributo “Estado” de la plantilla, los valores pueden ser: propuesto, revisado, validado, etc.

Tabla 24 Instancia de la clase Finito y de la clase ValorFinito

- InfinitoDefinido**. Representado como subclase de la clase **Manual**. Representa a un conjunto de valores infinitos dentro de un rango o límites específicos. Los atributos **LímSuperior** y **LímInferior** permiten especificar los límites respectivos que definen el rango de valores.

En la Tabla 25, mostramos, como ejemplo, a una instancia de la subclase **InfinitoDefinido**, asumido para la configuración de un atributo “Temperatura corporal”.

Clase	Atributo	Valor de ejemplo	Descripción
Naturaleza / Manual / Infinito Definido	LímSuperior	“50”	Para un atributo “Temperatura corporal” podemos especificar valores de 10 grados como límite inferior y 50 grados como límite superior.
	LímInferior	“10”	

Tabla 25 Instancia de la clase InfinitoDefinido

- InfinitoIndefinido**. Representado como subclase de la clase **Manual**. Representa a un conjunto de valores infinitos. El atributo **Descripción** de esta subclase puede recoger las características de estos valores.

En la Tabla 26, mostramos, como ejemplo, a una instancia de la subclase **InfinitoIndefinido**, asumido para la configuración de un atributo “Nombre del socio”.

Clase	Atributo	Valor de ejemplo	Descripción
Naturaleza / Manual / Infinito Indefinido	Descripción	“Miguel Grau Seminario”	Para un atributo “Nombre del socio” (nombre y apellidos comunes de personas), los valores de nombres son infinitos.

Tabla 26 Instancia de la clase InfinitoIndefinido

La clase **Relevancia**. Representa otra de las propiedades de configuración del atributo de la plantilla. Define la importancia o necesidad de especificación del valor del atributo de la plantilla. Esta característica permite definir en el sistema automático, que el atributo de la plantilla de especificación de requisito debe tener obligatoriamente un valor. Esta clase consta del siguiente atributo:

- Obligatorio**. Atributo booleano, que permite especificar el carácter obligatorio del valor del atributo para que proceda el registro del requisito. Por ejemplo, para la especificación de cualquier requisito, el atributo **Descripción** (atributo de la plantilla de especificación del requisito) es de suma importancia, por lo tanto, el atributo

## SOLUCIÓN PROPUESTA

**Obligatorio** (de la clase **Relevancia**) debe tomar el valor: “Sí”. Esto impide que el sistema informático registre al requisito hasta que sea especificada la descripción del requisito en un proyecto determinado.

En la Tabla 27, mostramos, como ejemplo, a una instancia de la clase **Relevancia**.

Clase	Atributo	Valor de ejemplo	Descripción
Relevancia	Obligatorio	“Sí”	El valor “Sí” impide el registro del requisito hasta que sea llenado el atributo de la plantilla respectivo al momento de su uso en un proyecto determinado.

*Tabla 27 Instancia de la clase Relevancia*

Clase **TipoDato**. Representa el tipo de dato que se podrá registrar en el atributo de la plantilla del requisito. Esta clase consta del siguiente atributo:

- **Tipo**. Indica al sistema automático el tipo de dato que será asignado al atributo de la plantilla del requisito.

En la Tabla 28, mostramos, como ejemplo, a una instancia de la clase **TipoDato**.

Clase	Atributo	Valor de ejemplo	Descripción
TipoDato	Tipo	“Fecha (Date)”	Define el tipo de dato del atributo de la plantilla de especificación de requisito. Puede ser de tipo: Texto, Carácter, Fecha, Booleano, etc.

*Tabla 28 Instancia de la clase TipoDato*

Clase **Control**. Otra de las propiedades de configuración del atributo. Permite definir el tipo de control automático para el registro del valor del atributo de la plantilla del requisito. Esta característica indica al sistema automático que tipo de control asignar al atributo específico para configurar la plantilla. Esta clase consta del siguiente atributo:

- **Control**. Indica al sistema automático el tipo de control que será asignado al atributo específico del requisito para la generación de la plantilla correspondiente.

En la Tabla 29, mostramos, como ejemplo, a una instancia de la clase **Control**.

Clase	Atributo	Valor de ejemplo	Descripción
Control	Control	“Listbox”	Tipo de control que se usará en la plantilla para registrar el valor del atributo. Pueden ser: Label, Buttonm Textbox, Radio button, Combobox, Listbox, etc.

*Tabla 29 Instancia de la clase Control*

Para que el ingeniero de software genere la plantilla de especificación de requisitos, tiene que especificar antes los valores correspondientes del modelo descrito. Con la información del modelo en el sistema informático, el ingeniero puede generar la plantilla de especificación del requisito. Una vez que el ingeniero de software ha generado la plantilla de requisitos, ya se pueden especificar los valores propiamente dichos del requisito reusable.

### 3.2.1.2 Asociaciones entre requisitos

En todo proyecto de software, los requisitos, que representan a las funcionalidades y restricciones del sistema, siempre tendrán algún tipo de relación entre sí. Estas relaciones se dan de manera natural dentro de todo sistema informático por diferentes

## SOLUCIÓN PROPUESTA

motivos: dependencias, necesidad, limitación u otro tipo de relación de un requisito respecto a otro.

El nivel de activo reusable de requisito individual de MORORE representa a un requisito en forma individual. Pero el modelo también propone la posibilidad de sugerir otros requisitos individuales que se encuentren asociados al requisito que se quiere reusar. Es decir, al momento que se selecciona un requisito para ser reusado, el sistema informático debe sugerir los requisitos que tengan algún tipo de relación dependiente con el requisito seleccionado para su posible reúso.

La clase-asociación **Asociación**. Representa la relación entre un requisito con otro que tenga algún tipo de relación. Un requisito reusable podrá tener o no dependencia con otros requisitos reusables almacenados.

En la Tabla 30, mostramos, como ejemplo, a una instancia de la clase-asociación **Asociación**.

Clase	Atributo	Valor de ejemplo	Descripción
Asociación	Tipo	“Restricción”	Por ejemplo, en forma general, el requisito “Registro de factura” puede tener relacionado al requisito no funcional “Los valores monetarios deben ser especificados siempre con dos decimales”. En este caso el atributo “Tipo” representa una asociación de tipo “restricción”. En este caso el requisito “Registro de factura” es el requisito dependiente del requisito “Los valores monetarios deben ser especificados siempre con dos decimales”.

*Tabla 30 Instancia de la clase-asociación Asociación*

### 3.2.1.3 Elementos relacionados

La definición de requisitos es un proceso creativo y evolutivo. En este proceso interactúan tanto los clientes como los desarrolladores. En muchos casos, para el desarrollo de los requisitos, se analizan políticas, normas, y otro tipo de referencias que toman un carácter relevante en el proceso de creación. MORORE permite la representación de elementos relacionados al requisito reusable. Estos elementos son la especificación de un conjunto de referencias que sirven para el entendimiento del requisito reusable.

A continuación, describimos al elemento del modelo que forma parte de esta característica del patrón de requisitos de MORORE:

La clase **ElementoRelacionado**. Representa a la referencia relevante que puede tener un requisito reusable. Esta clase consta de los siguientes atributos:

- **Nombre.** Representa al nombre del elemento relacionado al requisito instanciado. El elemento debe contener un nombre mnemónico, que sea fácil de recordar y que especifique en forma clara y concreta al elemento.
- **Descripción.** Contiene la información necesaria para entender o conocer al elemento relacionado.

## SOLUCIÓN PROPUESTA

- **Tipo.** Atributo que representa a los posibles tipos de referencia que se puede asignar al elemento relacionado.

En la Tabla 31, mostramos, como ejemplo, a una instancia de la clase **ElementoRelacionado**.

Clase	Atributo	Valor de ejemplo	Descripción
ElementoRelacionado	Nombre	“Reglamento de seguridad RS004 de la organización.”	Nombre del documento que hace referencia al elemento relacionado.
	Descripción	“El documento especifica las características de acceso al sistema informático...”	Descripción general para conocer o entender al elemento relacionado.
	Tipo	“Documento de la organización.”	Describe el tipo de elemento al que pertenece el elemento. Pueden ser: “página WEB, “Documento de la organización”, etc.

*Tabla 31 Instancia de la clase ElementoRelacionado*

### 3.2.1.4 Adaptación de los requisitos

El reúso lo podemos definir como el volver usar algo en un nuevo proyecto con mínimos cambios. En el proceso de reúso, los elementos reusables tienen que ser adaptados al nuevo proyecto. La adaptación consiste en cambiar los fragmentos particulares del elemento reusable por partes específicas del nuevo proyecto.

En este sentido, MORORE permite la asignación de parámetros a partes personalizables del valor de un atributo de un requisito reusable. El propósito de esta asignación de parámetros es facilitar al ingeniero la adaptación del requisito al nuevo proyecto.

Estos parámetros se especifican como variables que al momento de ser reusados tomarán un valor particular relacionado al proyecto donde se utilicen. El sistema informático, al momento de ser seleccionado el requisito para ser reusado, solicitará la asignación de los valores particulares relacionados al nuevo proyecto.

La clase **Parámetro**. Representa a la variable insertada en el valor del atributo del requisito. Esta variable se particulariza (adquiere un valor) al momento de reusar el requisito en un proyecto específico. Esta clase consta de los siguientes atributos:

- **Nombre.** Representa al nombre del parámetro asociado al valor de un atributo del requisito reusable. Por ejemplo, para un requisito que define la cantidad de tipos de usuario, el nombre de la variable es: “#TiposUsuarios”, representa la cantidad de tipos de usuarios que hay que llenar para el proyecto nuevo.
- **Posición.** Contiene la posición del parámetro dentro del valor del atributo del requisito.
- **Valor.** Representa el valor que adquiere el parámetro al momento de ser particularizado para ser reusado en un proyecto específico.

## SOLUCIÓN PROPUESTA

La clase **Parámetro**, está asociada a la clase **TipoDato**. Esta última, ha sido definida al final del acápite “3.2.1.1. Plantilla de requisitos configurable”. La clase **TipoDato** representan el tipo de dato que tomará el parámetro al ser reusado. Por ejemplo, para el parámetro “#TiposUsuarios” que define la cantidad de tipos de usuario, el tipo de dato del parámetro debe ser “Numérico”.

En la Tabla 32, mostramos, como ejemplo, a una instancia de la clase **Parámetro** y de la clase **TipoDato**, por ejemplo, para un atributo “Descripción” de un requisito, donde se define en el contenido la cantidad de tipos de usuario de un proyecto.

Clase	Atributo	Valor de ejemplo	Descripción
Parámetro	Nombre	“#TiposUsuarios”	Nombre del parámetro insertado en el valor del atributo del requisito reusable.
	Posición	35	Representa que el valor que se instancie, aparece a partir del carácter 35 del contenido del atributo del requisito.
	Valor	5	representa que en el nuevo proyecto habrán 5 tipos de usuarios.
TipoDato	Tipo	“Numérico”	Representa que el valor que tomará el parámetro debe ser numérico.

*Tabla 32 Instancia de la clase Parámetro y TipoDato*

### 3.2.2 Nivel de patrón de requisitos

El ingeniero de software se enfrenta día a día con una multitud de problemas de distintas magnitudes. Las mejores prácticas en ingeniería aconsejan “no reinventar la rueda”. Si continuamente se presentan problemas comunes, un buen ingeniero probará la efectividad de soluciones previas, volviendo a usar una misma solución adaptada para resolver problemas similares.

Como ya mencionamos anteriormente, la idea de patrones pretende reusar las experiencias en la solución de problemas recurrentes. El patrón permite hacer explícito el conocimiento de un experto en cuanto a la solución de un problema, para que pueda ser usado por otros en problemas similares. El presente trabajo de investigación toma la idea de patrón y la lleva al ámbito de la ingeniería de requisitos.

El segundo nivel de MORORE define el patrón de requisitos. Este patrón encierra un grupo de requisitos individuales relacionados, que son una solución efectiva a un problema recurrente en la especificación de software. El patrón de requisitos describe claramente al conjunto de requisitos solución, al problema al que hace referencia y además las distintas características de uso del patrón.

Al igual que en el nivel de requisito individual, la descripción del patrón de requisitos está orientado a su implementación en un sistema informático. En la Figura 11, mediante el diagrama de clases UML, mostramos los elementos que componen la representación del nivel de patrón de requisitos para MORORE.

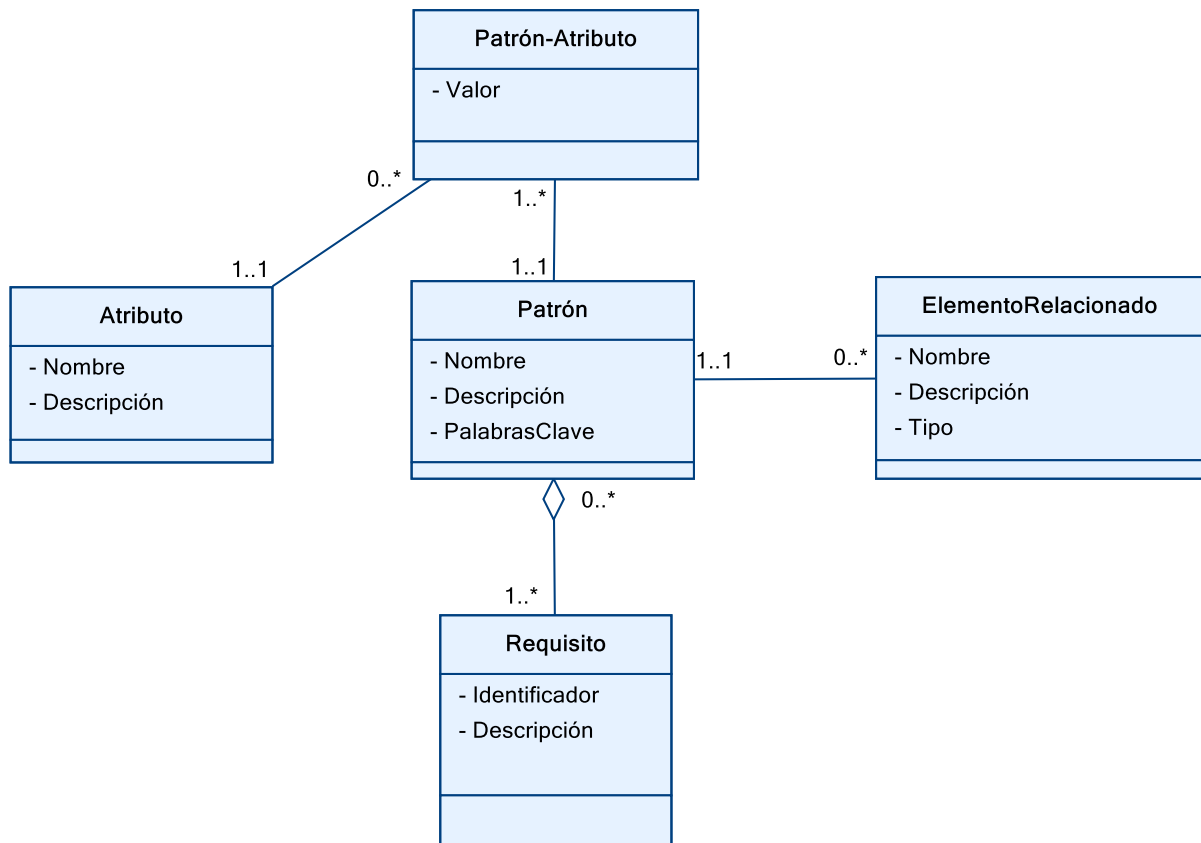


Figura 11 Modelo del nivel de patrón de requisitos de MORORE

Al igual que la descripción del nivel de requisito individual, el modelo de patrón de requisitos está descrito en áreas funcionales para una mejor comprensión del modelo general. Estas áreas funcionales son las siguientes:

### 3.2.2.1 Plantilla del patrón de requisitos

Para que los patrones puedan ser reusados, requieren ser guardados de tal manera que faciliten su posterior búsqueda, selección y adaptación a otros proyectos. En este sentido, MORORE plantea la creación de una plantilla específica para los patrones de requisitos.

Esta plantilla consta de un conjunto de atributos que permitan una descripción detallada y organizada del patrón de requisitos. Esta descripción debe permitir entender o conocer al patrón de requisitos para que pueda ser reusado por otros ingenieros de una forma eficiente.

Los elementos del nivel de patrón que componen la plantilla específica de descripción del patrón, los describimos a continuación:

La clase **Patrón** es el elemento que representa al patrón de requisitos. Por medio de esta clase representamos el conocimiento necesario para facilitar la indexación y recuperación del patrón. A continuación, describimos los atributos que brindan la información general y rápida para el conocimiento y uso del patrón.

## SOLUCIÓN PROPUESTA

- **Nombre.** Representa al nombre corto del patrón de requisitos para un mejor entendimiento del mismo al momento de su recuperación.
- **Descripción.** Permite describir las características más resaltantes del patrón de requisitos. Este atributo debe contener información clara y concreta que permita conocer que representa el patrón de requisitos.
- **PalabrasClave.** Contiene un conjunto de palabras independientes que facilitan la búsqueda del patrón de requisitos. Se recomienda que se utilicen entre tres y cinco frases separadas por comas.

En la Tabla 33, mostramos, como ejemplo, a una instancia de la clase **Patrón**.

Clase	Atributo	Valor de ejemplo	Descripción
Patrón	Nombre	“Patrón de acceso a un sistema de nivel 2”	Nombre corto para una rápida ubicación.
	Descripción	“Este patrón especifica los requisitos para definir la seguridad en el acceso a un sistema de información. Este grado de acceso debe considerar las restricciones del nivel 2 según la clasificación de dificultad de la empresa Asyncronyx”	Descripción general para facilitar el conocimiento del patrón.
	PalabrasClave	“Acceso, Nivel 2, Asyncronyx, Seguridad”	Palabras para facilitar la indexación y recuperación del patrón.

*Tabla 33 Instancia de la clase Patrón*

La clase **Atributo**. Representa a cada atributo descriptivo de la plantilla del patrón de requisitos. Este atributo constituye una característica específica del patrón de requisitos. El conjunto de las características describe en forma clara y concreta al patrón de requisitos.

En la Figura 11 apreciamos que la clase **Atributo** se relaciona con la clase **Patrón** por medio de la clase intermedia **Patrón-Atributo**, que describimos más adelante. A diferencia del nivel de requisito individual, el patrón de requisitos para MORORE consta de una única plantilla descriptiva para cualquier tipo de patrón. Los atributos que conforman la clase **Atributo** los especificamos a continuación:

- **Nombre.** Representa al nombre del atributo de la plantilla del patrón. Este elemento debe contener un nombre mnemónico, que sea fácil de recordar y que defina en forma clara y concreta al atributo. Se recomienda que el nombre no exceda las cinco palabras.
- **Descripción.** Contiene la información necesaria para entender o conocer al atributo genérico del patrón de requisitos.

En la Tabla 34, mostramos, como ejemplo, a una instancia de la clase **Atributo**.

## SOLUCIÓN PROPUESTA

Clase	Atributo	Valor de ejemplo	Descripción
Atributo	Nombre	“Ventajas y desventajas”	Nombre del atributo de la plantilla del patrón. Pueden ser: “Nombre”, “Clasificación”, “Ventajas y desventajas”, etc.
	Descripción	“Este atributo contendrá la información de los beneficios y perjuicios que se pueden dar al usar el patrón”	Descripción general del atributo de la plantilla del patrón.

Tabla 34 Instancia de la clase Atributo

Para definir la plantilla de patrón de requisitos de MORORE, realizamos una revisión bibliográfica y de estándares relacionados. Al final obtuvimos un conjunto de atributos, que consideramos permiten describir al patrón en forma suficiente. Estos atributos los describimos en la Tabla 35.

Atributo	Descripción
Identificador	Identifica unívocamente al patrón.
Clasificación	Clasifica al patrón de requisitos.
Nombre	Nombre corto del patrón de requisitos.
Descripción	Descripción detallada del patrón de requisitos.
Palabras claves	Palabras relacionadas que permitan una búsqueda eficiente del patrón de requisitos.
Autor	Nombre del creador del patrón de requisitos.
Ventajas	Descripción de las bondades puede brindar el uso del patrón de requisitos.
Desventajas	Descripción de los perjuicios que puede causar el uso del patrón de requisitos.
Condiciones	Descripción de las situaciones específicas para el uso del patrón de requisitos.

Tabla 35 Atributos para la plantilla del patrón de requisitos de MORORE

La clase **Patrón-Atributo**. Es la clase intermedia entre la clase **Patrón** y la clase **Atributo**. Esta clase representa al valor que toma cada atributo de la plantilla de un patrón. Esta clase consta del atributo siguiente:

- **Valor**. Representa al valor que se instanciará para un atributo que pertenezca a un patrón.

En la Tabla 36, mostramos, como ejemplo, a una instancia de la clase **Patrón-Atributo**.

Clase	Atributo	Valor de ejemplo	Descripción
Patrón-Atributo	Valor	“Seguridad”	Representa el valor que adquiere un atributo de la plantilla del patrón. Para el ejemplo, se asume el valor de un atributo “Clasificación”. Los valores pueden ser: “Registro de datos”, “Seguridad”, etc.

Tabla 36 Instancia de la clase intermedia Patrón-Atributo

### 3.2.2.2 Elementos relacionados

En forma análoga a la característica del nivel de requisito individual, el patrón en su proceso de creación o identificación pudo requerir el análisis de políticas, normas, y otro tipo de referencias que toman un carácter relevante. MORORE permite representar



## SOLUCIÓN PROPUESTA

estos elementos relevantes que sirven como referencia para el entendimiento, en este caso, del patrón de requisitos.

A continuación, describimos al elemento del modelo que forma parte de esta característica del patrón de requisitos de MORORE:

La clase **ElementoRelacionado**. Esta clase representa la descripción de la referencia relevante que puede tener un patrón de requisitos. Esta clase consta de los siguientes atributos:

- **Nombre.** Representa al nombre del elemento relacionado al patrón de requisitos. El elemento debe contener un nombre mnemónico, que sea fácil de recordar y que identifique claramente al elemento relacionado.
- **Descripción.** Especifica en forma detallada la información para entender o conocer al elemento relacionado.
- **Tipo.** Define los posibles tipos de elementos relacionados al patrón de requisitos.

En la Tabla 37, mostramos, como ejemplo, a una instancia de la clase **ElementoRelacionado**.

Clase	Atributo	Valor de ejemplo	Descripción
ElementoRelacionado	Nombre	“ISO/IEC 27001, estándar para la seguridad de la información”	Representa el nombre del elemento relacionado al patrón.
	Descripción	“El documento especifica las características técnicas de acceso al sistema informático”	Representa la descripción del elemento relacionado.
	Tipo	“Estándar”	Representa el tipo de elemento relacionado. Pueden ser: “Página web”, “Documento de la organización”, “Estándar”, etc.

*Tabla 37 Instancia de la clase ElementoRelacionado*

### 3.2.2.3 Conjunto de requisitos solución

La plantilla del patrón de requisitos, descrita anteriormente (Tabla 35), define el conjunto de requisitos como grupo. En esta sección del modelo, representamos la descripción de los requisitos que componen al patrón. Cabe señalar que la siguiente descripción del requisito del patrón puede asumir, si es el caso, todas las características descritas en la sección “3.2.1 Nivel de requisito individual”.

A continuación, especificamos el elemento del modelo que representa esta descripción de requisitos del patrón:

La clase **“Requisito”**. Esta clase permite representar a los requisitos que forman parte del patrón. El conjunto de estos requisitos conforma la solución al problema recurrente que define al patrón. Para que un requisito forme parte de un patrón debió ser parte de la solución original de donde se obtuvo el patrón y contribuir relevantemente a la solución del problema. Los atributos que conforman la clase **“Requisito”** los describimos a continuación:

## SOLUCIÓN PROPUESTA

- **Identificador.** Representa un valor que identifica unívocamente al requisito que conforma el patrón. Por ejemplo: “R001”.

**Descripción.** Es el atributo donde se especifica al requisito. Este atributo se usa para describir detalladamente al requisito que forma parte del patrón. Es importante que la descripción sea clara y fácil de entender, se deben evitar términos técnicos y ambiguos. Por ejemplo: “el software debe permitir definir los usuarios que tendrán acceso al sistema...”. En la Tabla 38, mostramos, como ejemplo, a una instancia de la clase **Requisito**.

Clase	Atributo	Valor de ejemplo	Descripción
Requisito	Identificador	“R001”	Representa el identificador unívoco del requisito.
	Descripción	“el software debe permitir definir los usuarios que tendrán acceso al sistema...”.	Representa la descripción detallada del requisito.

*Tabla 38 Instancia de la clase Requisitos*

### 3.2.3 Nivel de estructura de tipos de requisitos

Como hemos mencionado, los principales problemas que afronta la ingeniería de software tienen su origen en la ingeniería de requisitos, en forma específica en una deficiente especificación y gestión de requisitos (Ibañez & Rempp, 1996).

Como una alternativa de mejorar la especificación, muchos autores y estándares proponen distintas formas de organizar los requisitos. Los propios ingenieros, por experiencia, crean sus propias maneras de clasificación de requisitos en dependencia de las características del software a desarrollar: tipo, magnitud, cliente u organización, personal involucrado, etc.

El tercer nivel de MORORE define el patrón de estructuras de tipos. Este patrón permite contener un conjunto de clasificaciones de requisitos organizados en una estructura común. El patrón de estructuras de tipos de MORORE soporta las diferentes estructuras de clasificaciones propuestas por los autores ( (Braude & Bernstein, 2016), (Pressman & Maxim, 2015), (Sommerville, 2016)) y estándares revisados en la presente investigación ( (CC, 2017), (ESA, 2017), (ISO/IEC/IEEE-29148, 2017), (Metrica-3, 2017)). Además, el modelo de patrón de estructuras cuenta con otras características adicionales que permiten un mayor alcance de estructuras de tipos de requisitos (asociaciones entre tipos, modelado recursivo de los tipos, etc.).

En la Figura 12 representamos el nivel de patrón de estructuras de tipos de requisitos mediante el diagrama de clases UML correspondiente (Hurtado, et al., 2006). Al igual que los anteriores niveles, orientamos la descripción del patrón de estructuras a su implementación en un sistema informático.

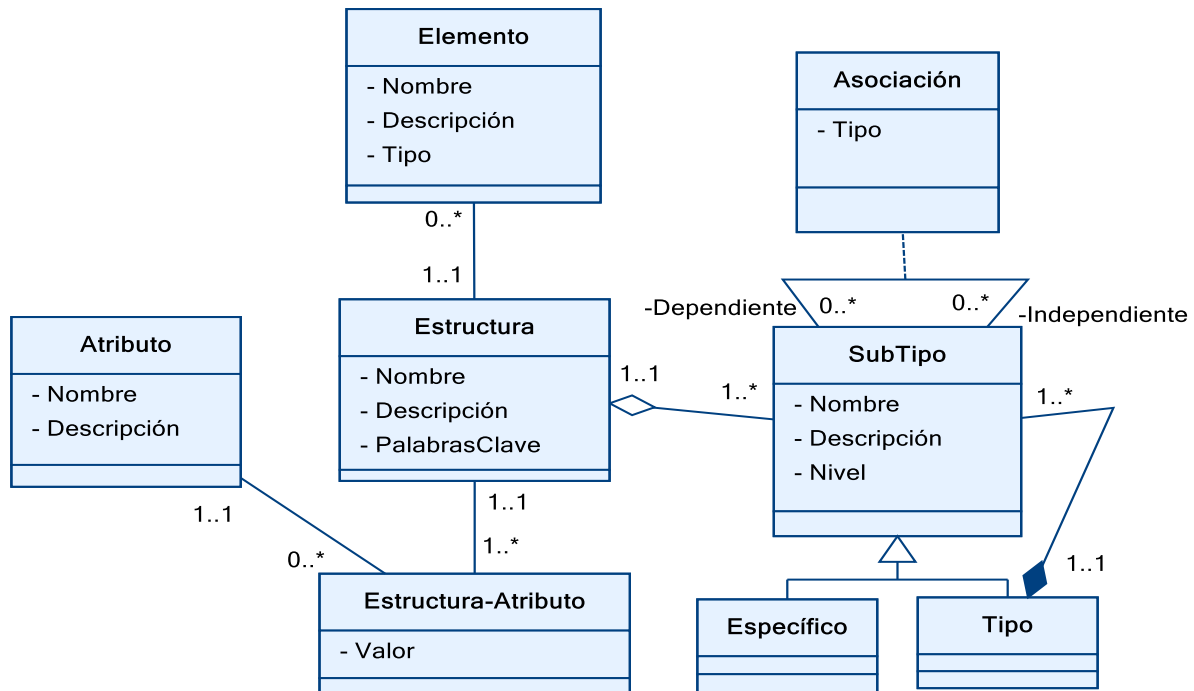


Figura 12 Modelo del nivel de patrón de estructura de tipos de requisitos de MORORE

Similar a la descripción del nivel de requisito individual y patrón de requisitos, el modelo de patrón de estructuras de tipos está descrito en áreas funcionales para una mejor comprensión del modelo general.

### 3.2.3.1 Plantilla del patrón de estructuras de tipos

Tomando la idea de patrón, en igual forma que para el patrón de requisitos, surge la necesidad de hacer explícito el conocimiento tácito de la experiencia de expertos en cuanto a organización de requisitos. Todo esto con el objetivo de que este conocimiento pueda ser reusado por profesionales nóveles.

Igualmente, los patrones de estructuras de tipos requieren ser guardados de tal manera que faciliten su posterior reúso. En este sentido MORORE plantea la creación de una plantilla específica para describir los patrones de estructuras de tipos requisitos.

La plantilla consta de un conjunto de atributos que permitan una descripción detallada y organizada del patrón de estructuras de tipos de requisitos. Esta descripción debe permitir entender o conocer al patrón de requisitos para que pueda ser reusado por otros ingenieros de una forma eficiente.

Los elementos del nivel de patrón de estructuras, que componen la plantilla específica de descripción del patrón, los describimos a continuación:

La clase **Estructura** es el elemento que representa al patrón de estructuras de tipos general. Esta clase permite especificar el conocimiento explícito acerca de la estructura, con el propósito de hacer eficiente su indexación y recuperación. A continuación, describimos los atributos que brindan la información general y rápida para el conocimiento y uso del patrón de estructuras:

## SOLUCIÓN PROPUESTA

- **Nombre.** Representa al nombre corto del patrón de estructuras de tipos para un mejor entendimiento del mismo al momento de su recuperación.
- **Descripción.** Permite describir las características más resaltantes del patrón de estructuras de tipos. Este atributo debe contener información clara y concreta que permita conocer que representa el patrón de estructuras de tipos.
- **PalabrasClave.** Contiene un conjunto de palabras independientes que facilitan la búsqueda del patrón de estructuras de tipos. Se recomienda que se utilicen entre tres y cinco frases separadas por comas.

En la Tabla 39, mostramos, como ejemplo, a una instancia de la clase **Estructura**.

Clase	Atributo	Valor de ejemplo	Descripción
Estructura	Nombre	“Estructura del estándar de la ESA”	Nombre corto para una rápida ubicación.
	Descripción	“Clasificación de requisitos del estándar de la Agencia Espacial Europea: Guide to the user requirements definition phase. And, ESA PSS-05-03 Issue 1 Revision 1: Guide to the software...”.	Descripción general para facilitar el conocimiento de la estructura.
	PalabrasClave	“Estándar de la ESA, Organización de requisitos, estructura de tipos de requisitos, etc.”	Palabras para facilitar la indexación y recuperación de la estructura.

*Tabla 39 Instancia de la clase Estructura*

La clase **Atributo**. Representa a cada atributo descriptivo del patrón de estructuras de tipos. Este atributo constituye una característica específica del patrón de estructuras. El conjunto de características describe en forma clara y concreta al patrón de estructuras de tipos.

En la Figura 12, apreciamos que la clase **Atributo** se relaciona con la clase **Estructura** por medio de la clase intermedia **Patrón-Atributo**, que describimos a continuación. A diferencia del nivel de requisito individual, el patrón de estructuras para MORORE consta de una única plantilla descriptiva para cualquier tipo de patrón.

Los atributos que conforman la clase **Atributo** los especificamos a continuación:

- **Nombre.** Representa al nombre del atributo de la plantilla del patrón. Este elemento debe contener un nombre mnemónico, que sea fácil de recordar y que defina en forma clara y concreta al atributo. Se recomienda que el nombre no exceda las cinco palabras.
- **Descripción.** Contiene la información necesaria para entender o conocer al atributo genérico del patrón de estructuras.

## SOLUCIÓN PROPUESTA

En la Tabla 40, mostramos, como ejemplo, a una instancia de la clase **Atributo**.

Clase	Atributo	Valor de ejemplo	Descripción
Atributo	Nombre	“Ventajas y desventajas”	Nombre del atributo de la plantilla del patrón. Pueden ser: “Nombre”, “Descripción”, “Ventajas y desventajas”, etc.
	Descripción	“Este atributo contendrá la información de los beneficios y perjuicios que se pueden dar al usar la estructura”	Descripción general del atributo de la plantilla del patrón de estructuras.

*Tabla 40 Instancia de la clase Atributo*

Del mismo modo que los niveles anteriores, para definir la plantilla de patrón de estructuras de tipos de MORORE, realizamos una revisión bibliográfica y de estándares relacionados. Obtuvimos un conjunto de atributos, que consideramos permiten describir al patrón en forma suficiente. Estos atributos los describimos en la Tabla 41.

Atributo	Descripción
Identificador	Identifica unívocamente al patrón.
Clasificación	Clasifica al patrón de requisitos.
Nombre	Nombre corto del patrón.
Descripción	Descripción detallada del patrón.
Palabras clave	Palabras relacionadas que permitan una búsqueda eficiente del patrón de estructuras.
Autor	Nombre del creador del patrón.
Ventajas	Descripción de las bondades que puede brindar el uso del patrón de estructuras de tipos.
Desventajas	Descripción de los perjuicios que puede causar el uso del patrón de estructuras de tipos.
Condiciones	Descripción de las situaciones específicas para el uso del patrón de estructuras.

*Tabla 41 Atributos para la plantilla del patrón de estructuras de tipos*

La clase **Estructura-Atributo**. Es la clase intermedia entre la clase **Estructura** y la clase **Atributo**. Esta clase representa al valor que toma cada tipo de atributo de la plantilla del patrón. Esta clase consta del atributo siguiente:

- **Valor.** Representa al valor que se instanciará para un atributo de la plantilla del patrón.

En la Tabla 42, mostramos, como ejemplo, una instancia de la clase **Estructura-Atributo**.

Clase	Atributo	Valor de ejemplo	Descripción
Estructura-Atributo	Valor	“Organización de la ESA”	Ejemplo para un atributo “Nombre” de la plantilla del patrón.

*Tabla 42 Instancia de la clase Estructura-Atributo*

### 3.2.3.2 Elementos relacionados

En forma análoga a la característica del nivel de requisito individual y patrón de requisitos, el patrón de estructuras, en su proceso de creación o identificación, pudo requerir el análisis de estándares, libros y otro tipo de referencias que toman un carácter relevante. MORORE permite representar estos elementos relevantes que sirven como referencia para el entendimiento del patrón de estructuras de tipos.

## SOLUCIÓN PROPUESTA

A continuación, describimos al elemento del modelo que forma parte de esta característica del patrón de estructuras de tipos de requisitos de MORORE:

La clase **ElementoRelacionado**. Esta clase representa la descripción de la referencia relevante que puede tener un patrón de estructuras. Esta clase consta de los siguientes atributos:

- **Nombre.** Representa al nombre del elemento relacionado al patrón de estructuras. El elemento debe contener un nombre mnemónico, que sea fácil de recordar y que identifique claramente al elemento relacionado.
- **Descripción.** Especifica en forma detallada la información para entender o conocer al elemento relacionado.
- **Tipo.** Define los posibles tipos de elementos relacionados al patrón.

En la Tabla 43, mostramos, como ejemplo, a una instancia de la clase **ElementoRelacionado**.

Clase	Atributo	Valor de ejemplo	Descripción
ElementoRelacionado	Nombre	“Estándar de Ingeniería de Software de la ESA”	Representa el nombre del elemento relacionado al patrón.
	Descripción	“El documento es el contenido del estándar de la ESA, específicamente en su guía de la fase de ingeniería de requisitos...”	Representa la descripción del elemento relacionado.
	Tipo	“Estándar”	Representa el tipo de elemento relacionado. Pueden ser: “Página web”, “Documento de la organización”, “Estándar”, etc.

Tabla 43 Instancia de la clase ElementoRelacionado

### 3.2.3.3 Estructura de tipos de requisitos: modelo recursivo

Los autores, estándares, los ingenieros experimentados, etc. proponen muchas formas de organizar requisitos. Estas formas pueden ser muy variadas en nombres, estructura, niveles, relaciones, etc. El tercer nivel de MORORE, que define al patrón de estructuras, permite soportar la representación de todas las estructuras encontradas en la revisión de la literatura realizada en la presente tesis doctoral. Es más, se añadieron algunas características propias para aumentar la cobertura de este nivel.

En este apartado se representa a la estructura de tipos y subtipos descrita en forma general por la clase **Estructura**. Me diante el diagrama de clases de la Figura 12, representamos la estructura de tipos y subtipos empleando la técnica de recursividad usando herencia y composición de UML.

Con la técnica de recursividad podemos representar una estructura de árbol de cualquier nivel de profundidad. Esto nos permite representar al conjunto de tipos y subtipos en forma jerárquica de una estructura de tipos de requisitos.

## SOLUCIÓN PROPUESTA

A continuación, especificamos cada uno de los elementos que componen la parte recursiva del modelo del nivel de estructuras de tipos:

La clase **SubTipo** está asociada a la clase **Estructura**. Esta clase **Subtipo**, se constituye como clase abstracta, que generaliza a las subclases **Tipo** y **Específico**, que representan a los tipos de requisitos. Esta clase consta de los siguientes atributos:

- **Nombre.** Atributo que representa al nombre del tipo de clasificación requisitos de la estructura. El tipo debe contener un nombre mnemónico, que sea fácil de recordar y que identifique claramente al tipo respectivo.
- **Descripción.** Atributo que especifica, en forma detallada, información para conocer al tipo de clasificación de requisitos, es decir, describe al tipo de requisito que puede contener.
- **Nivel.** Atributo que especifica el nivel jerárquico del tipo de clasificación instanciado, dentro de la estructura de tipos específica.

La subclase **Tipo**. Permite representar a la clasificación específica de requisitos propiamente dichos. Pero este tipo obligatoriamente contiene subtipos de clasificaciones en forma recursiva. Para poder representar esta recursividad, a pesar de que la clase **Tipo** es subclase de la clase abstracta **SubTipo**, la clase **Tipo** mantiene una asociación de composición con la clase **SubTipo**. Este tipo “No Funcional” puede contener a otros subtipos: No funcional de restricción, No funcional de capacidad, No funcional de seguridad, etc.

La subclase **Específico**. Permite representar una hoja del árbol, es decir, que representa a los tipos de requisitos finales o que ya no tiene subtipos. Esta subclase permite que la recursividad no se convierta en un ciclo infinito. Por ejemplo: el tipo “No funcional de capacidad” que ya contendría a los requisitos de capacidad y que no tiene subtipos.

En la Tabla 44, mostramos, como ejemplo, a una instancia de la clase **Tipo** o también pudo ser de la clase **Específico**. Ambas clases tienen los mismos atributos, la diferencia es que una tiene subtipos (**Tipo**) y la otra no (**Específico**).

Clase	Atributo	Valor de ejemplo	Descripción
SubTipo: Tipo (puede tener subtipos) o Específico (No puede tener subtipos)	Nombre	“No funcional”	Representa el nombre de un tipo de requisito de la estructura.
	Descripción	“Permite contener a los requisitos que son restricciones o limitaciones de otros requisitos...”	Describe al tipo de requisito que contendrá una clasificación de la estructura.
	Nivel	“2”	Representa que la clasificación en referencia está en el segundo nivel de la estructura.

Tabla 44 Instancia de la clase Tipo

### 3.2.3.4 Asociaciones entre tipos de clasificaciones de requisitos

Como mencionamos, se realizó el análisis de muchas estructuras de tipos. En algunas estructuras notamos que algunos tipos podían tener relaciones con otros tipos de



## SOLUCIÓN PROPUESTA

requisitos en una misma estructura. Por ejemplo: Los requisitos funcionales se obtiene en base a requisitos de usuario de capacidad, según el estándar de la ESA (ESA, 2017).

El nivel de patrón de estructuras de tipos de requisitos de MORORE, brinda la posibilidad de representar asociaciones entre los tipos de clasificaciones de una misma estructura de tipos.

La clase asociación: **Asociación**. Representa la relación entre tipos de clasificaciones de requisitos de una misma estructura. Permite que un tipo de clasificación tenga algún tipo de relación con otro tipo de clasificación dentro de la misma estructura de tipos de requisitos. El tipo de asociación esta descrito en el atributo **Tipo**.

En la Tabla 45, mostramos, como ejemplo, a una instancia de la clase asociación: **Asociación**.

Clase	Atributo	Valor de ejemplo	Descripción
Asociación	Tipo	“Dependencia”	Por ejemplo para el tipo de requisito de software “Funcional”, <i>depende</i> del tipo de requisito de usuario de “Capacidad”.

Tabla 45 Instancia de la clase asociación: Asociación

### 3.3 Modelo de calidad

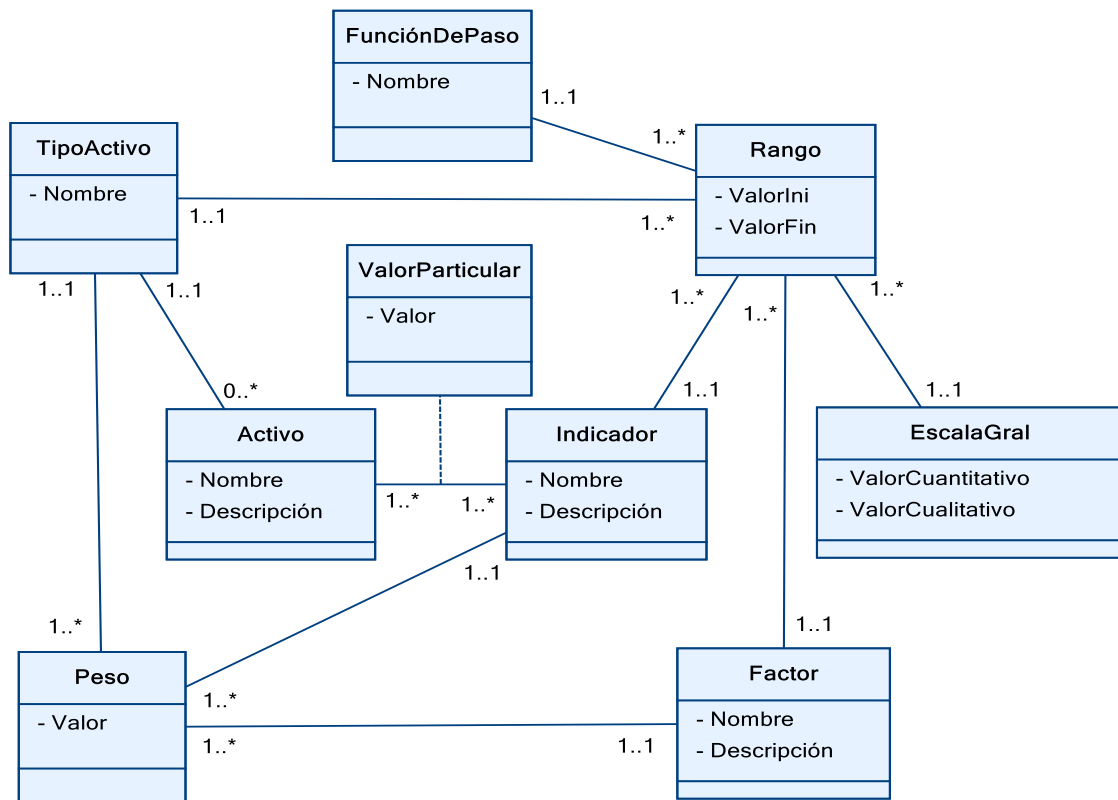
La presente sección de la tesis doctoral, está basada en dos trabajos de investigación “*A framework to measure and improve the quality of textual requirements*” (Génova, et al., 2013) y “*Modelo para medir la calidad de activos de la ingeniería de requisitos*” (Sánchez, et al., 2015), en los cuales el autor de la presente tesis doctoral es coautor.

Como mencionamos anteriormente, la importancia de obtener alta calidad en la ingeniería de requisitos, es un factor primario para el éxito de un proceso de desarrollo de software. Sin los requisitos bien definidos y especificados, un proyecto de desarrollo de software está destinado a fracasar.

Muchos estándares ( (ESA, 2017), (IEEE, 1990), (Metrica-3, 2017)) y autores ( (Braude & Bernstein, 2016), (Loucopoulos & Karakostas, 1995), (Sommerville, 2016), (Thayer & Dorfman, 1997) (Génova, et al., 2013), (Marrero, et al., 2008)) coinciden en que los requisitos deben ser claros, exactos, completos, etc. Pero a pesar de que se habla de la calidad en el proceso de software y de la importancia de la ingeniería de requisitos, se ha trabajado poco en establecer medios formales de control de calidad para los requisitos. Pues las cualidades que pueden determinar la calidad de los requisitos, en la práctica son difíciles de medir y no son tomados en cuenta en muchos casos.

Entonces ¿cómo hacemos efectivo el control de calidad en el ámbito de la ingeniería de requisitos? En la presente tesis doctoral planteamos un modelo de evaluación de la calidad de activos software del ámbito de la ingeniería de requisitos. Este modelo se muestra en la Figura 13. Específicamente, el modelo, está orientado a evaluar la calidad de los activos definidos en MORORE. Pero los principios que hemos desarrollado, bien se pueden trasladar a otro tipo de activos del proceso de desarrollo de software en general.





*Figura 13 Modelo de Calidad de activos de requisitos de MORORE*

En MORORE planteamos que, para aplicar el control de calidad de los activos, se debe definir una combinación de cualidades deseables a las que llamamos “factores de calidad”. Estos factores son cuantificables mediante un sistema de métricas formado por elementos definidos para el modelo. Estos elementos permiten una medición efectiva y práctica para cada factor de cada activo definido en MORORE.

A continuación describimos los elementos que conforman el modelo de calidad de MORORE (Figura 13). Dentro de un proceso de medición de la calidad, estos elementos del modelo de calidad permiten identificar a los activos de requisitos factibles de mejorar su especificación.

### 3.3.1 Factores de calidad

Definimos el “factor de calidad” de MORORE como una cualidad apreciable en un requisito individual, patrón de requisitos o patrón de estructura de tipos. Esta cualidad permite valorar como buena o mala la calidad de los activos software. Podemos mencionar que en la literatura revisada existen distintas opiniones acerca de qué cualidades deseables deben de tener los requisitos de software. Sin embargo, respecto a cualidades acerca de los patrones de requisitos y patrones de estructuras de tipos, no encontramos mucha información. Por consiguiente, la definición de factores está basada principalmente en la literatura revisada y en el análisis respectivo de los activos software definidos para MORORE.

## SOLUCIÓN PROPUESTA

---

Asimismo, para especificar y clasificar a los factores de calidad de MORORE, también tomamos en cuenta a los activos de requisitos en estudio, principalmente por su respectivo fin u objetivo:

- El objetivo de los requisitos en forma individual es especificar un sistema informático.
- El objetivo del patrón de requisitos es brindar una solución a problemas comunes de especificación que se presentan en el ámbito de la ingeniería de requisitos. Y
- El objetivo del patrón de estructuras de tipos es organizar a los requisitos para una mejor especificación.

Otro punto a tomar en cuenta en la definición de los factores, son los interesados, principalmente los clientes e ingenieros. Aunque los interesados, conciben el proyecto de software desde distintos puntos de vista, ambos se orientan a obtener activos de requisitos de calidad. En este sentido, consideramos necesario generalizar sus apreciaciones para lograr una mayor comprensión de los factores a describir:

- Desde el punto de vista del cliente, los activos de requisitos deben representar lo que el producto resultante hará o los problemas que solucionará.
- Desde el punto de vista de los ingenieros, los activos de requisitos deben permitir construir correctamente al producto resultante y de esta manera permitir su eficiente mantenimiento.

Entonces, podemos resumir que para obtener la lista de cualidades que permiten analizar la calidad de los activos de requisitos principalmente utilizamos la combinación de un conjunto de elementos referentes. Estos elementos son:

- La revisión de la literatura relacionada. De la bibliografía revisada, obtuvimos algunas cualidades que tienen relación principalmente con los requisitos en forma individual.
- La naturaleza de los factores o cualidades deseables de los requisitos. En algunos casos el conocimiento de las propiedades encontradas en la literatura revisada, permitió generalizarlas y aplicarlas a los otros activos definidos en MORORE: patrones de requisitos y patrones de estructuras de tipos.
- La naturaleza de los activos de requisitos. Sobre todo, la utilidad del activo de requisitos definido, también ha permitido inferir los factores especificados para evaluar la calidad.
- El punto de vista de los interesados en la calidad de los activos. Las diferentes apreciaciones de los interesados respecto a los activos de requisitos, también colaboró en la definición de los factores de calidad. Los interesados son principalmente los que definen si los activos son buenos o malos.

A continuación, describimos la clase que representa a los factores de calidad del modelo.

La clase **Factor**. Representa a cada cualidad que permite evaluar la calidad del activo definido en MORORE. En la Figura 13, apreciamos que la clase “Factor” tiene los atributos que definen a cada cualidad y que especificamos a continuación:

## SOLUCIÓN PROPUESTA

---

- **Nombre.** Representa al nombre del factor de calidad. Este elemento debe contener un nombre mnemónico, que sea fácil de recordar y que defina en forma clara y concreta a la cualidad. Se recomienda que el nombre tenga una sola palabra. Por ejemplo: el factor: “Comprensibilidad”, “Validabilidad”, “Precisión”, etc.
- **Descripción.** Contiene la información necesaria para entender la cualidad que afecta a un activo. Por ejemplo, para un factor “Atomicidad” su descripción sería: “Este factor propugna que el activo esté claramente determinado e identificado. Cada activo debe ser visto como una unidad coherente sin que se mezcle con otros activos”.

A continuación, describimos los factores determinados para evaluar la calidad de los activos de requisitos de MORORE. La lista presentada en esencia está descrita de forma general. En las secciones posteriores del documento especificamos las relaciones de los factores respecto a los activos de requisitos. Hay que recalcar que la siguiente lista de factores es definida para evaluar al activo en forma independiente cara a su almacenamiento:

- **Validabilidad.** Este factor determina si el activo de requisitos expresa adecuadamente al sistema informático que satisfaga las necesidades del cliente.
- **Verificabilidad.** Este factor determina si el activo de requisitos tiene la capacidad de ser comprobado de manera objetivamente una vez que sea implementado.
- **Modificabilidad.** Este factor determina si el activo tiene las condiciones que permitan su fácil mantenimiento ante los frecuentes cambios que pueden sufrir durante todo el proceso de desarrollo de software. También tiene que ver con los futuros cambios del software ya realizado, pues si no es fácil modificar los requisitos tampoco será fácil modificar el software producido.
- **Integridad.** También encontrada en la literatura revisada como “Compleitud”. Esta cualidad hace referencia a que los elementos de un activo deben englobar todos los aspectos básicos y esenciales del problema o situación que se está tratando. Un activo debe cubrir en la medida de lo posible todo el espectro del problema o necesidad para lo que fue creado. Se debe tener especial cuidado en esta cualidad, pues por el afán de tener un activo completo podemos incluir aspectos que no son propios de la situación tratada, y se estaría afectando directamente otras cualidades como, por ejemplo, la atomicidad.
- **Consistencia.** También conocida como “Concordancia”. Este factor hace referencia a minimizar la posibilidad de encontrar contradicciones entre los elementos de un activo. Asimismo, relacionamos con esta cualidad el evitar las redundancias y solapamientos. Es decir, evitar los elementos funcional o estructuralmente iguales y minimizar las intersecciones entre los elementos del activo respectivamente. Por ejemplo, como mencionamos anteriormente que la descripción de un patrón debe ser coherente, ya que debe responder al problema que se plantea; de igual forma el conjunto de requisitos o la estructura que conforma el patrón debe ser concordante y no mostrar contradicciones. Dichos requisitos deben integrarse y relacionarse de tal forma que funcionen armónicamente. No se puede permitir que exista en un patrón en el que existan elementos que se opongan ya sea parcial o totalmente, pues esto va directamente en contra de la calidad.

## SOLUCIÓN PROPUESTA

---

- **Inambigüedad.** Es la cualidad por la cual la especificación de un activo no debe dar lugar a más de una interpretación. La descripción del activo debe ser suficientemente claro al momento de detallar lo que éste es capaz de hacer. Se debe tener cuidado en la utilización de términos o estructuras y toda clase de expresiones que puedan entenderse de distintos modos o aceptar más de una interpretación. Al evitar el uso de estos elementos ambiguos se ofrece una mejor comprensión del activo. Sin embargo, no se debe exagerar en la utilización de términos demasiado específicos respecto al dominio del problema, pues esto podría afectar la generalidad del patrón respecto a su reúso.
- **Atomicidad.** Este factor propugna que el activo esté claramente determinado e identificado. Cada activo debe ser visto como una unidad coherente sin que se mezcle con otros activos. En el caso del patrón responde a una situación o problema específico, aunque este puede presentarse en muchos entornos distintos. Al hablar de atomicidad, se habla de un conjunto integrado, y en el caso de un patrón, de un conjunto de requisitos que necesitan trabajar juntos ya que forman una combinación específica que responde a un problema o situación puntual planteada; siempre resaltando que puede pertenecer a muchos contextos distintos.
- **Abstracción.** El alcance de la ingeniería de requisitos, entre otras cosas, es especificar claramente lo que el producto software debe hacer, sin necesidad de establecer como realizarlo. Del mismo modo el activo debe de ser lo suficientemente abstracto para establecer las características del producto software, sin que entre en el detalle de cómo hacerlo. Se debe evitar el exceso de detalles técnicos en la especificación del activo.
- **Comprensibilidad.** Un activo de requisitos es comprensible en la medida que su especificación es entendida sin dificultad. Se debe buscar que esté bien redactado, especificando claramente su alcance. Para ser entendible no es necesario emplear términos de uso común, sin embargo, no se puede negar que un texto será más entendible en la medida que se utilice terminología propia de hablar cotidiano. Aún si se utiliza muchos términos propios de entorno o dominio en el que se desarrolla, puede ser entendible en cuánto se facilite al usuario un glosario de términos.
- **Precisión.** También conocida como “Concisión” o “Exactitud”. Esta cualidad busca que la especificación de un activo este bien orientada. El alcance de un activo debe ser concreto (bien detallado) y exacto. Esta cualidad puede confundirse con la ambigüedad, pero el activo podría tener una sola interpretación, pero no tener la orientación exacta que se requiere.
- **Frecuencia.** Este factor nos indica cuán común es una situación o cuán repetitivo es el uso de un activo. Así un activo será frecuente cuando detalle una situación o realidad recurrente a la cual se plantea solución. Al decir que un activo de requisitos es frecuente, estamos afirmando que el problema al que ha dado solución es específico, sin embargo, se presenta muchas veces y en múltiples entornos distintos.
- **Coherencia.** Es la cualidad por la que se establece una relación directa entre dos objetos o entidades. Al hablar de coherencia, estamos hablando que todos los elementos de un activo deben tener un orden lógico. Así, un activo es coherente cuando todos sus elementos están alineados a un mismo fin común. Por ejemplo, la descripción del patrón debe reflejar la coherencia del patrón al puntualizar una situación que corresponda al conjunto de requisitos o estructuras que conforman el patrón.

## SOLUCIÓN PROPUESTA

---

- **Correctitud.** por definición un activo será correcto cuando esté libre de errores. Tanto sus elementos como su estructura deben ser desarrollados de tal manera que sean lógicos. Esta cualidad es propia de cualquier activo de cualquier tipo, pues, una vez desarrollado el activo debe estar bien hecho para poder ser utilizado en otro proyecto. Con mayor razón, si se pretende utilizar un activo para reusarlo, hay que asegurarse que esté libre de cualquier tipo de error.
- **Sencillez.** para calificar un activo como sencillo se debe considerar el tipo de lenguaje que se utiliza; sobre todo en la descripción del mismo. Mientras menos palabras técnicas o específicas estén presentes en la especificación de un activo, mayor será la posibilidad de que el lector lo entienda. Esta cualidad está muy ligada a la cualidad de comprensibilidad, sin embargo, una descripción podría ser sencilla en cuanto a uso de términos, pero no comprensible dado que no se han usado, por ejemplo, las formas verbales apropiadas.
- **Adaptabilidad.** Esta cualidad se entiende como la capacidad de un elemento para modificar detalles de su disposición. El objetivo es adecuarse a un nuevo entorno en el que se desempeñará con un mínimo de modificaciones. Un activo de calidad se desarrolla dentro de un proyecto específico. Sin embargo, al existir la necesidad de reúso, el activo deberá ser capaz de aceptar cambios en la especificación sin dificultad. Y así adaptarse fácilmente de forma óptima en los nuevos proyectos en los que se le incluya. Esto no significa que el activo sea cambiado en su esencia, pues entonces ya sería un nuevo activo.
- **Simplicidad.** un activo es simple cuánto menos complejos sean los elementos que lo forman. Mientras menos partes y menos complejas sean las relaciones que existan entre una parte y otra, más simple es el activo. Sin embargo, tanto el número de partes como la forma en que se relacionen éstas son de suma importancia para que el activo se consolide como tal. Es necesario, en tal sentido, encontrar un equilibrio entre la interdependencia de las partes y la simplicidad del activo. Un activo demasiado complejo no se podrá reusar fácilmente puesto que su adaptación será complicada.
- **Organización:** este factor hace referencia al orden de los elementos dentro de la estructura del activo. Es importante un orden lógico, pues un activo debe entenderse fácilmente para que el proceso de reúso se lleve sin mayores inconvenientes.

### 3.3.2 Indicador de calidad

En la sección anterior especificamos los factores o cualidades que debe tener un activo de calidad para que pueda ser reusado efectivamente. En este apartado describimos el cómo medir las características deseables que identificamos.

Definimos un indicador de calidad como una métrica formalizada para cuantificar una cualidad. Como se sabe, las cualidades son apreciadas de manera subjetiva según lo que el observador esté buscando o el rol que desempeñe dentro del proyecto. Esta subjetividad no implica que sean arbitrarios, sino difícilmente cuantificables. En este sentido, necesitamos establecer un sistema de medida efectivo para evaluar las cualidades.

La clase **Indicador**. Representa a cada métrica que permite evaluar objetivamente a cada cualidad del activo definido en MORORE. En la Figura 13, apreciamos que la

## SOLUCIÓN PROPUESTA

clase “indicador” tiene los atributos que definen a cada métrica y que especificamos a continuación:

- **Nombre.** Representa al nombre del indicador. Este elemento debe contener un nombre mnemónico, que sea fácil de recordar y que defina en forma clara y concreta al indicador. Se recomienda que el nombre no supere las 10 palabras. Por ejemplo: el indicador: “Promedio de frases por párrafo”, “Número de términos ambiguos”, etc.
- **Descripción.** Contiene la información necesaria para entender al indicador. Por ejemplo, para un indicador “número de palabras” su descripción sería: “especifica la cantidad de palabras que debe tener un activo en su especificación. Esta métrica es un punto de referencia, permite determinar el tamaño que dará una idea más general de la dimensión del activo”.

Los indicadores, determinados en este modelo, no son arbitrarios ni suficientes para fijar si una cualidad está presente o no en un activo; sino que serán un punto de referencia para los ingenieros. Lo que se busca es aproximar y facilitar el trabajo. Siempre será necesario el juicio de quien utilice las métricas para llegar a alguna conclusión final.

Como los atributos de los activos típicamente son llenados en lenguaje natural, para el modelo principalmente escogimos algunos indicadores basados en el análisis de textos. Pero también podemos encontrar indicadores basados en la estructura misma de los activos, fundamentalmente para los patrones (componentes, relaciones, etc.).

Para una mejor descripción de los indicadores, los clasificamos basándonos principalmente en el análisis de textos, pues, estamos analizando activos descritos en lenguaje natural. Así, los indicadores desarrollados los clasificamos en cinco grupos principales, según su naturaleza mostrados en la Tabla 46.

Clasificación	Descripción
Indicadores morfológicos	Son aquellos indicadores que analizan un texto desde el punto de vista formal, sin tener en cuenta el contenido. Los indicadores morfológicos nos dan una idea del tamaño y de la composición del texto, sin tomar en cuenta su significado. Nos ayudarán a comprender el tamaño los atributos y su composición.
Indicadores léxicos	Son aquellos indicadores que miden propiedades relativas al contenido del texto. Nos ayudarán a analizar el vocabulario empleado y cómo este afecta a las propiedades del activo.
Indicadores analíticos	Son aquellos indicadores que estudian las propiedades del texto utilizando herramientas lingüísticas más complejas, haciendo hincapié en la naturaleza y categoría de las palabras y sus combinaciones.
Indicadores relacionales	Son aquellos indicadores que nos muestran las conexiones del activo con otros elementos, ya sean estructuras, páginas web, artículos, especificaciones, etc. o incluso otros activos. Asimismo también toma en cuenta las relaciones existentes entre los elementos internos del propio activo si es el caso.
Indicadores de desarrollo	Son los indicadores que nos manifiestan cómo ha evolucionado el activo. Examinan los cambios sufridos y las modificaciones necesarias para que el activo pueda ser reusado efectivamente.

*Tabla 46 Clasificación de los indicadores de calidad*

A continuación, describimos los indicadores que permiten medir con mayor objetividad a los factores o cualidades de calidad de los activos de MORORE. Al igual

## SOLUCIÓN PROPUESTA

---

que los factores, la siguiente descripción de los indicadores está realizada de forma general, independientemente del activo y la cualidad a la que afecten. En secciones posteriores describimos el detalle de las relaciones pertinentes respecto al modelo de calidad.

### 3.3.2.1 Indicadores morfológicos.

- **Tamaño.** El tamaño de la especificación se enfoca básicamente en las dimensiones del activo a analizar. El tamaño puede ser medido de distintas formas, por ejemplo, por el número de caracteres, de palabras, de frases, de párrafos, etc. En el modelo realizado, el tamaño de la especificación de los activos es medido en función de los indicadores siguientes:
  - **Número de palabras:** como el sistema de medida es un punto de referencia, escogimos el número de palabras para determinar el tamaño porque dará una idea más general de la dimensión del activo, y es una forma de medida más útil en orientación al objetivo buscado.
  - **Número de elementos de la solución:** este indicador está orientado a los patrones, tanto de requisitos como de estructuras de tipos de MORORE. Un patrón presenta un conjunto de componentes que forman una solución a un problema recurrente, que puede ser reusado muchas veces. En este sentido, subyacentemente, si la cantidad de elementos del conjunto solución es muy poca, no cubrirá una solución concreta coherente. Por otro lado, si la cantidad de elementos es muy grande, puede desvanecer y hacer poco reusable a la solución planteada.
- **Legibilidad lingüística.** Se entiende por legibilidad lingüística la adecuada redacción de un texto, de tal forma que sea fácil de leer. Existen diversos indicadores morfológicos para indicar el grado de legibilidad lingüística, por ejemplo: promedio de caracteres por palabra, promedio de sílabas por palabra, promedio de palabras por frase, promedio de frases por párrafo, etc. En general mientras más simples o reducidos sean los elementos del texto de especificación del activo a analizar, más legible será ya que implicará menos esfuerzo al entenderlo. Para este modelo se realizará la medición de la legibilidad en función de:
  - **Promedio de sílabas por palabra:** la idea de que mientras más corta es una palabra es más fácil de entender en el conocimiento general. En el hablar cotidiano se evita el uso de palabras excesivamente largas, y por el contrario se utiliza sinónimos más cortos, por ejemplo, en vez de decir “Estoy en mi domicilio” decimos “Estoy en mi casa”. O en vez de decir “Escapó el presidiario” se suele decir “Escapó el reo”.
  - **Promedio de palabras por oración:** así como es más legible una palabra de menor cantidad de sílabas, una frase u oración será más legible en cuánto posea menos palabras, sobre todo en los atributos de un patrón en los que se busca precisión. Por ejemplo, en vez de decir “Es una persona de corta edad” se suele decir “Es una persona joven”. O en vez de decir “Es una persona que sigue cursos en la universidad” se dice “Es un estudiante universitario”.
  - **Promedio de frases u oraciones por párrafo:** es la misma idea de los dos indicadores anteriores, aunque a un nivel de abstracción diferente. Un párrafo

## SOLUCIÓN PROPUESTA

---

debe representar una idea, pero esta idea debe estar sustentada por algunas premisas, y si es el caso, mostrar algunas consecuencias. Cada uno de estos elementos los representamos por medio de una frase u oración. En este sentido, si excedemos el número de premisas o consecuencias, aumenta el riesgo de hacer menos entendible a la idea principal.

- **Puntuación.** La puntuación hace referencia a los signos necesarios para hacer más entendible una frase. También es necesaria cuándo se desea expresar ideas relacionadas pero que no se complementan de tal manera que puedan expresarse en una sola oración. Sin embargo, el excesivo uso de los signos de puntuación o la ausencia de los mismos puede hacer un texto más difícil de leer y de entender. Los signos de puntuación precisan usarse en la medida justa y cuando sean estrictamente necesarios. El indicador usado para el modelo es:
  - Promedio de signos de puntuación por frase u oración: como está descrito, este indicador hace referencia al uso óptimo (ni poco ni mucho) de los signos de puntuación por frase.
- **Acrónimos y abreviaturas.** Las siglas (CASE, ESA, etc.) y las abreviaturas (Doc., Dr., etc.) son simplificaciones de texto que pueden afectar el entendimiento de la especificación del activo. El indicador usado en el modelo dentro de esta clasificación es:
  - Número de acrónimos o abreviaturas: existen acrónimos o abreviaturas de uso frecuente o que pueden estar bien definidas en el glosario del documento. Pero el abuso de estos elementos aumenta la posibilidad de perjudicar la calidad de los activos analizados.

### 3.3.2.2 Indicadores léxicos

Como ya mencionamos, son conjuntos de palabras que pueden afectar las propiedades del activo. Estos indicadores los clasificamos de la siguiente manera:

- **Términos conectivos.** Son elementos de texto que permiten cualquier construcción lingüística. No obstante, el hacer un uso excesivo de estos términos resulta pernicioso para la calidad del activo. En base a este tipo de términos definimos los siguientes indicadores:
  - Número de conjunciones copulativas-disyuntivos: una conjunción se define como la palabra que sirve de nexo entre dos oraciones. Estas conjunciones pueden ser de distintos tipos, sin embargo, el indicador se centra en dos tipos: número de conjunciones copulativas y número de conjunciones disyuntivas. Las conjunciones copulativas (“y”, “e”) son aquellas que sirven para expresar en unidad dos o más elementos homogéneos. Por otro lado, las conjunciones disyuntivas (“o”, “u”) expresan alternativa. Cuando utilizamos conjunciones lo que estamos haciendo es “mezclar” dos oraciones en una sola, lo cual puede crear problemas de claridad y legibilidad al momento de analizar un texto. Es preferible utilizar dos oraciones cortas en vez de una sola oración más larga, por lo cual el uso de conjunciones, ya sea copulativas o disyuntivas, debe hacerse cuando sea estrictamente necesario.
  - Número de términos negativos: estos términos denotan negación de algo y hacer que la especificación sea difícil de comprender. Además, pueden aumentar el



## SOLUCIÓN PROPUESTA

---

riesgo de generar inconsistencias lógicas en el texto. Estos términos pueden ser “no”, “ni”, “nunca”, “nada”, “ninguna” etc.

- Número de términos de flujo de control: son términos que pueden afectar el entendimiento del flujo de control de una función o proceso, y sobrepasar los límites de especificación que deben mostrar los requisitos (el “QUÉ” debe hacer el software y no el “CÓMO”). Entre estos términos tenemos, por ejemplo: “mientras”, “cuando”, “si... entonces”, etc.
- Número de términos anafóricos: hace referencia a los términos capaces de sustituir a otros términos. Estos términos incrementan el riesgo de imprecisiones y ambigüedades en el texto de especificación. En este grupo podemos encontrar a los pronombres personales (“él”, “ello”), pronombres relativos (“que”, “donde”), los pronombres demostrativos (“éste”, “ése”, “aqué”), etc.
- Términos imprecisos. Estos términos por su misma naturaleza denotan ambigüedad al texto de especificación de un activo. Específicamente son palabras que no dan precisión hacia un objetivo. Especificamos a los siguientes indicadores de acuerdo al tipo de imprecisión que pueden generar:
  - Número de términos imprecisos de calidad: “bueno”, “malo”, “adecuado”, “eficiente”, etc.
  - Número de términos imprecisos de cantidad: “bastante”, “poco”, “aproximadamente”, “algo”, etc.
  - Número de términos imprecisos de frecuencia: “casi siempre”, “generalmente”, “usualmente”, etc.
  - Número de términos imprecisos de enumeración: “en un futuro”, “no limitado a”, “varios”, etc.
  - Número de términos imprecisos de probabilidad: “posiblemente”, “probablemente”, “opcionalmente”, etc.
  - Número de términos imprecisos de usabilidad: “adaptable”, “extensible”, “fácil”, “familiar”, “seguro”, etc.
- Términos de diseño. Son los términos que hacen referencia al diseño del software (el “CÓMO”) o a la tecnología. Estos términos afectan a la abstracción de los requisitos, los cuales típicamente deben describir “QUÉ” debe hacer el software, mas no, el “CÓMO” hacerlo. El indicador obtenido de este grupo es el siguiente:
  - Número de términos de diseño: “parámetro”, “MySQL”, “Java”, “bucle”, etc.

Los indicadores descritos en la presente sección son fácilmente identificables por sus propiedades características. Pero su comportamiento respecto al modelo de calidad de la presente tesis es similar. Por este motivo los indicadores de esta sección, que usamos para evaluar los factores, los usamos agrupados en su nivel general:

- Número de términos conectivos.
- Número de términos imprecisos.
- Número de términos de diseño.

## SOLUCIÓN PROPUESTA

---

### 3.3.2.3 Indicadores analíticos.

Por su naturaleza, estos indicadores requieren un análisis de texto con herramientas lingüísticas relativamente complejas. Para el presente trabajo utilizamos los siguientes indicadores pertenecientes a este grupo:

- Número de errores de ortografía y gramática: todo texto entendible necesita, por lo menos, el uso de palabras correctamente escritas y conjugadas. Eso implica tener especial cuidado en no incluir errores de ortografía o gramática (“pérdida - perdida”, “número - numeró”, “no, es verdad - no es verdad”, etc.) que puedan influir en la inteligibilidad del texto; aunque el no hacerlo tampoco asegura que el texto sea descifrable. Aún más tratándose de un documento de especificación de requisitos, que es el nexo entre los clientes y los ingenieros, y sobre el cual se desarrollará todo el proyecto.
- Número de términos ambiguos: llamamos términos ambiguos a aquellos que admiten dos o más significados distintos (“Luna: satélite o apellido”, “banco: entidad financiera o mueble o conjunto de peces”, etc.). Estos términos pueden crear confusión pues al entenderse de varios modos generan dudas razonables. En la especificación de un activo debemos evitar que los términos usados sean ambiguos pues, como ya se mencionó, el activo formará parte de la especificación de requisitos que es la base del proyecto de software. Podemos, hasta cierto punto, evitar las dudas y confusiones si anexamos un glosario de términos y definiciones que permita aclarar cualquier inquietud sobre algún significado.
- Número de formas verbales imperativas y número de formas verbales indicativas: el verbo constituye una parte importante de la oración pues nos permite identificar “qué” se hace, y dependiendo del tiempo y modo verbales, “cuándo” se hace. La especificación de los activos debe ser lo más sencilla posible, desde el punto de vista de la redacción, para que así sea correctamente comprendido. Es recomendable, por tanto, utilizar formas verbales simples que disminuyan la posibilidad de dudas en el lector. En este sentido los modos indicativos (“visualizará”, “permite”, “escribo”, etc.) e imperativos (“escribe”, “pulsa”, “digita”, etc.), expresan acciones u órdenes concretas respectivamente. En este sentido, la ausencia de estos modos verbales podría disminuir la precisión en la descripción, afectando también a la atomicidad. En forma similar, el abuso de estos modos verbales en un activo sería contraproducente. Por lo tanto, es recomendable el uso apropiado (ni mucho, ni poco) de estos modos verbales para mejorar la calidad de los activos.
- Número de formas verbales condicionales y número de formas verbales compuestas: como mencionamos en la sección anterior, la especificación de los activos debe ser lo más sencilla posible. Por lo tanto, se recomienda evitar las formas verbales condicionales (“abriría”, “debería”, etc.), pues estas hablan de casos hipotéticos, mientras lo que se busca en un activo es precisión. Lo mismo pasa con las formas verbales compuestas (“habría podido”, “habría realizado”, etc.), se deben usar cuando es estrictamente necesario.
- Número de términos de dominio: se llama términos de dominio al conjunto de palabras o expresiones propias de un entorno específico. Por ejemplo, las palabras “requisito”, “UML”, “lenguajes de programación” y “caso de uso” son términos de dominio de la Ingeniería de Software, así como “bisturí”, “porta agujas” y “pinzas de disección” son términos de dominio de la cirugía convencional. Dentro del ámbito de

los requisitos para el reúso, no conviene exagerar el uso de términos del dominio del proyecto. Así, si el patrón se ha desarrollado dentro de un proyecto de software para una ferretería, el patrón que se almacenará para el reúso debe contener más términos generales que términos del dominio de la ferretería, pues así facilita la adecuación del patrón a un entorno distinto a aquel en el que se desarrolló.

### 3.3.2.4 Indicadores relacionales.

Los indicadores relacionales tienen como función analizar las relaciones existentes entre los elementos de un mismo activo, así como las relaciones con otros elementos del entorno. En este sentido, para el presente trabajo definimos los siguientes indicadores:

- Número de relaciones con otros elementos: estas relaciones hacen referencia a las distintas conexiones que pueden existir entre el activo y otro tipo de elemento de su entorno u otro activo de su misma naturaleza. De esta manera, el indicador busca medir la cantidad de elementos (páginas WEB, documentos, otros activos, etc.) que tienen relación con el activo.
- Número de dependencias entre elementos del activo: en el acápite anterior hicimos referencia a las relaciones externas del activo, pero también se debe tomar en cuenta las relaciones internas. Los elementos que forman un activo, sobre todo los patrones de requisitos y estructuras de tipos, están relacionados unos con otros y se complementan entre sí. Estos vínculos pueden ser pocos y sencillos, o muchos y complejos. De cualquier forma, estos vínculos, nos brindan información sobre la estructura del patrón y por tanto nos ayudan a analizar su comportamiento y cualidades.
- Número de solapamientos: la relación de solapamiento se manifiesta entre activos que tienen partes funcionales o algunas características estructurales comunes entre sí.
- Número de niveles de anidamiento: cuanto más complejo es un activo se hace más necesario utilizar más niveles de anidamiento, sin embargo, no se debe llegar a la exageración pues resultaría contraproducente. Si se establecen demasiados niveles de anidamiento por tratar de mantener un mejor orden, se podría complicar la estructura del activo innecesariamente.

### 3.3.2.5 Indicadores de desarrollo

Los indicadores de desarrollo buscan dar información adicional acerca de cómo se comporta el activo durante el proceso de creación de mismo y acerca de las experiencias de uso, sobre todo las positivas. De este grupo obtenemos los siguientes indicadores:

- Número de versiones: las versiones de un activo surgen cuando se le hacen pequeñas variantes para adaptarlo a un grupo de situaciones más específicas. Cuando hablamos de “pequeñas variantes”, se refiere a que los cambios no afectan el núcleo del activo, ni su naturaleza, pues en ese caso ya estaríamos tratando otro activo distinto al original.
- Número de éxitos: este indicador mide cuántas veces ha sido utilizado el activo de forma exitosa. Para considerar un “éxito” no basta con que el activo haya sido incluido

## SOLUCIÓN PROPUESTA

en un proyecto de software, sino que este debe haber sido validado y se debe haber comprobado que, dentro del proyecto, el activo ha sido utilizado sin problemas.

### 3.3.3 Activo de requisitos

Continuando la descripción del modelo de calidad, encontramos a los activos. Los activos, son los elementos reusables definidos para MORORE: requisito individual, patrón de requisitos y patrón de estructuras de tipos.

La clase **Activo**. Representa al activo específico que puede ser reusado. En la Figura 13, apreciamos que la clase “Activo” tiene el siguiente atributo:

- **Nombre.** Representa el nombre corto del activo para un mayor entendimiento del mismo. Por ejemplo: “Cantidad de niveles de acceso de un software”.
- **Descripción.** Contiene la información necesaria para entender al activo. Por ejemplo, para un tipo de activo “requisito individual” su descripción sería: “Especifica la cantidad de niveles de acceso que debe tener el software X”.

La clase **Tipo de activo**. Representa a los tres tipos de activos definidos para MORORE: requisito individual, patrón de requisitos y patrón de estructuras de tipos. El atributo “**Nombre**” permite especificar el nombre del tipo de activo en cuestión. Los nombres de los tipos de activos son: “Requisitos individual”, “Patrón de requisitos” y “Patrón de estructuras de tipos”.

La clase **Peso**. Ya se han definido los factores, indicadores y tipos de activos. En base a estos elementos se define el “peso”. El “peso” representa la ponderación o importancia que tiene un indicador respecto a otro dentro del grupo de indicadores que permiten evaluar a un “Activo” en función a un “Factor” y “Tipo de activo”. El atributo “Valor” especifica el peso respectivo de un “indicador”, para la evaluación de un “activo” dependiendo del “tipo” y respecto a un “factor”.

Por ejemplo: para evaluar la “inambigüedad” de un “requisito individual”, asumamos que lo hacemos por medio de los siguientes indicadores: “Número de términos Imprecisos (NTI)”, “Número de términos conectivos (NTC)” y “Número de fallos ortográficos o gramaticales (NFO)”. Como la evaluación es respecto a la “inambigüedad” de un requisito, la importancia de los indicadores debe ser definida en base a que indicador afecta más al nivel de inambigüedad del texto. En este sentido determinamos que el indicador que menos afecta, de los 3 definidos, es el NTI, mientras los que más afectan son: NTC y NFO. Y consideramos que para la inambigüedad de un texto el NFO es más importante que el NTC. La Tabla 47 muestra los pesos específicos de los indicadores del ejemplo.

Indicadores para la Inambigüedad de un Requisito	Peso
NFO (Número de fallos ortográficos y gramaticales)	4
NTC (Número de términos conectivos)	3
NTI (Número de términos imprecisos)	2

*Tabla 47 Ejemplo del peso ponderado asignado a indicadores respecto a un Factor de un Activo*

### 3.3.4 Escalas de las métricas

En las secciones anteriores describimos los factores que determinan la calidad de los activos. Estos factores son difíciles de medir, por lo cual también describimos un conjunto de indicadores por cada factor de cada tipo de activo. Sin embargo, no todos los indicadores afectan a cada uno de los factores. Dependiendo de su naturaleza, cada factor tiene un grupo de indicadores determinantes. De esta forma, un factor puede poseer un sólo indicador, o muchos de ellos. Así mismo, un indicador se puede relacionar con un sólo factor o con muchos de ellos. La forma en que se relacionan factores e indicadores la presentamos en las secciones subsiguientes.

Ahora bien, al manejar diferentes indicadores para un mismo factor, y cada factor tiene naturaleza propia, surge un problema: a distintos indicadores les corresponde distintas unidades de medición. Estas particularidades de cada indicador, requirieron un análisis en áreas más específicas de acuerdo a la naturaleza del indicador. Al valor específico de la medida propia de un indicador en un activo concreto, le denominamos “Valor particular”.

La clase **ValorParticular**. Es la clase asociación entre las clases **Activo** y la clase **Indicador**. Esta clase representa el valor particular o la medida que tiene el activo específico al momento que es evaluado respecto a un indicador determinado. Esta clase consta del atributo **Valor**. Este atributo representa al valor de la medida del activo específico respecto a un indicador determinado. Por ejemplo, para el activo “patrón de seguridad de EAL2 (EAL, Evaluation Assurance Levels: Nivel de Confianza de la Evaluación) del Common Criteria”, se mide respecto al indicador “promedio de palabras por oración”, la medida sale “16.58” que se asigna al atributo **Valor**”.

Continuando con la descripción del modelo, conocemos que el valor particular de un indicador puede tener unidad de medición distinta al que tiene otro indicador, por lo tanto, no son comparables en un mismo sistema de medición. Sin embargo, para salvar este problema, podemos definir una escala general que permita unificar las mediciones de los distintos indicadores.

Planteamos el siguiente ejemplo a fin de aclarar el concepto: para el factor “Adaptabilidad” de un “Requisito”, se establece un indicador “número de palabras (NP)” y un indicador “número de términos de diseño (NTD)”. Mientras el NP se maneja dentro de una unidad de medida que va de 1 a 100, el indicador NTD se maneja dentro de una unidad de medida que va de 1 a 10. Al analizar un activo determinado, resulta que el indicador NP marca 20 y el indicador NTD marca 5. A partir de aquí surgen las preguntas: ¿puedo promediar estos valores?, ¿qué me dicen estos valores respecto al factor? ¿los valores son buenos o malos? Respecto a la pregunta si ¿podemos promediar estos valores?, definitivamente no. Antes tenemos que unificar las medidas en lo que ya denominamos “Escala General”.

La clase **EscalaGral**. Esta clase representa a una unidad de medida genérica en la cual confluyen las unidades de los distintos indicadores en forma unificada. Para nuestro modelo de calidad, asumimos que la medida particular de un indicador, corresponda con uno de tres posibles rangos relacionados al indicador respectivo según su unidad de medida. Estos tres rangos se corresponden con las tres escalas generales respectivamente.

## SOLUCIÓN PROPUESTA

Las tres escalas generales que se consideran en nuestro modelo son: “bueno”, “medio” o “malo”. Por ejemplo, para el factor “inambigüedad” de un “requisito”, el indicador de “cantidad de términos imprecisos (CTI)”, se considera “bueno” si el valor de medida es 0 o máximo 1, “medio” si la medida es 2 y “malo” si la medida es de 3 a más. Y si a nuestra escala cualitativa le asignamos valores numéricos, tendremos a la vez una escala cuantitativa que nos permita un mejor manejo en el proceso de la evaluación de calidad de los activos de requisitos. En la Tabla 48, mostramos los valores del ejemplo descrito, con sus respectivas escalas generales tanto cualitativa como cuantitativa.

Rango de CTI respecto a la inambigüedad de un requisito	Escala General	
	Cualitativa	Cuantitativa
De 0 a 1	Bueno	1
De 2 a 2	Medio	2
De 3 a más	Malo	3

*Tabla 48 Ejemplo de la aplicación de la escala de un indicador a la Escala General*

Los atributos de la clase “**EscalaGral**” que representan respectivamente a la escala cualitativa y escala cuantitativa respectivamente son:

- **ValorCuantitativo.** Represente al valor cuantitativo de la Escala General: 1 para bueno, 2 para medio y 3 para malo.
- **ValorCualitativo.** Represente al valor cualitativo de la Escala General. “Bueno” (1), “Medio” (2) y “Malo” (3).

Tenemos que hacer algunas puntualizaciones. Primero que escogimos para la escala general un sistema de 3 valores para unificar los sistemas de medida de todos los indicadores. Consideramos que estos tres valores son suficientes para comparar y definir la calidad de los activos de requisitos de MORORE. Otro aspecto que puntualizar es que para el modelo de MORORE, la escala general cualitativa y cuantitativa se corresponde unívocamente, es decir, siempre para el valor “bueno” le corresponde el valor 1, para el valor “medio” le corresponde el valor 2 y para el valor “malo” le corresponde el valor 3.

### 3.3.5 Función de paso

Con la “escala general” resolvemos el problema de la unificación de los sistemas de medida de los distintos indicadores. Pero aún quedaron pendientes las preguntas, que ante un valor en las unidades del sistema de medida del indicador particular ¿Qué nos dicen respecto al factor?, ¿el valor es bueno o malo? Para solucionar este problema, planteamos que un rango de valores de un indicador en su sistema particular de medida, le corresponde una única escala general dependiendo de la relación entre el indicador y el factor respectivo de un tipo de activo concreto.

Para comprender mejor este aspecto planteamos un ejemplo con el indicador “cantidad de términos imprecisos (CTI)” respecto al factor “inambigüedad” de un requisito. Subyacentemente encontramos que a mayor CTI disminuye la calidad y a menor CTI aumenta la calidad del requisito respecto al factor “inambigüedad”. Esto denota una tendencia decreciente del requisito respecto al factor de calidad. Ahora, también podemos encontrar otro ejemplo, el indicador de tamaño “número de palabras” respecto al factor “atomicidad” de un requisito. Aunque este indicador no es muy determinante, encontramos que a menor número de palabras la calidad baja respecto al

## SOLUCIÓN PROPUESTA

---

factor “atomicidad”. Pues, el factor “Atomicidad” propugna que el requisito esté claramente identificado, pocas palabras podrían no describir bien al requisito. Pero, un excesivo número de palabras también disminuye la calidad. Esto último debido a que el un excesivo número de palabras puede hacer perder la especificidad del requisito. Sin embargo, conocemos que la especificación óptima de un requisito debe ser ni muy reducida ni muy grande, es decir, es buena cuando la cantidad de palabras de su especificación tiene valores intermedios. Esta relación sigue una tendencia “convexa”.

La clase **FunciónDePaso**. Representa la relación de transformación que tiene la medida del indicador respecto al factor del tipo de activo respectivo, con la escala general por medio del rango específico. La utilidad de esta función de paso es transformar la medida numérica particular de un indicador de calidad y homogenizar medidas de diferentes indicadores respecto a un mismo factor de calidad para un tipo de activo. Por los ejemplos descritos, podemos mencionar que la forma en que cada factor de un activo se relaciona con cada uno de sus indicadores varía. Por ejemplo, en algunos casos, una medida alta del indicador será mejor para el factor, mientras que en otros casos una medida baja del indicador será lo deseable. Incluso puede ocurrir que lo ideal en un indicador no sea lo más alto o lo más bajo, sino que se desee que el indicador se encuentre en un rango intermedio determinado y viceversa. Por consiguiente, dependiendo cómo se correspondan los factores e indicadores, los tipos de relación pueden ser de cuatro tipos principales, mostrados en la Figura 14:

- Función de paso creciente. Una función de paso es creciente siempre y cuando, ante una medida alta del indicador, el factor se califica mejor, es decir, cuanto más alto sea la medida del indicador la calidad respecto al factor es mejor.
- Función de paso decreciente. La función de paso decreciente se da siempre y cuando, ante una medida alta del indicador, el factor se califica como peor, es decir, cuanto más alto sea la medida del indicador, la calidad respecto al factor es peor.
- Función de paso convexa. La función de paso convexa se da siempre y cuando, ante una medida intermedia del indicador, el factor se califica como alto y para las medidas extremas (pequeñas o grandes) el factor se califica como bajo.
- Función de paso cóncava. La función de paso cóncava es contraria a la convexa, se da siempre y cuando, ante una medida intermedia del indicador, el factor se califica como bajo y para las medidas extremas (pequeñas o grandes) el factor se califica como alto.

## SOLUCIÓN PROPUESTA

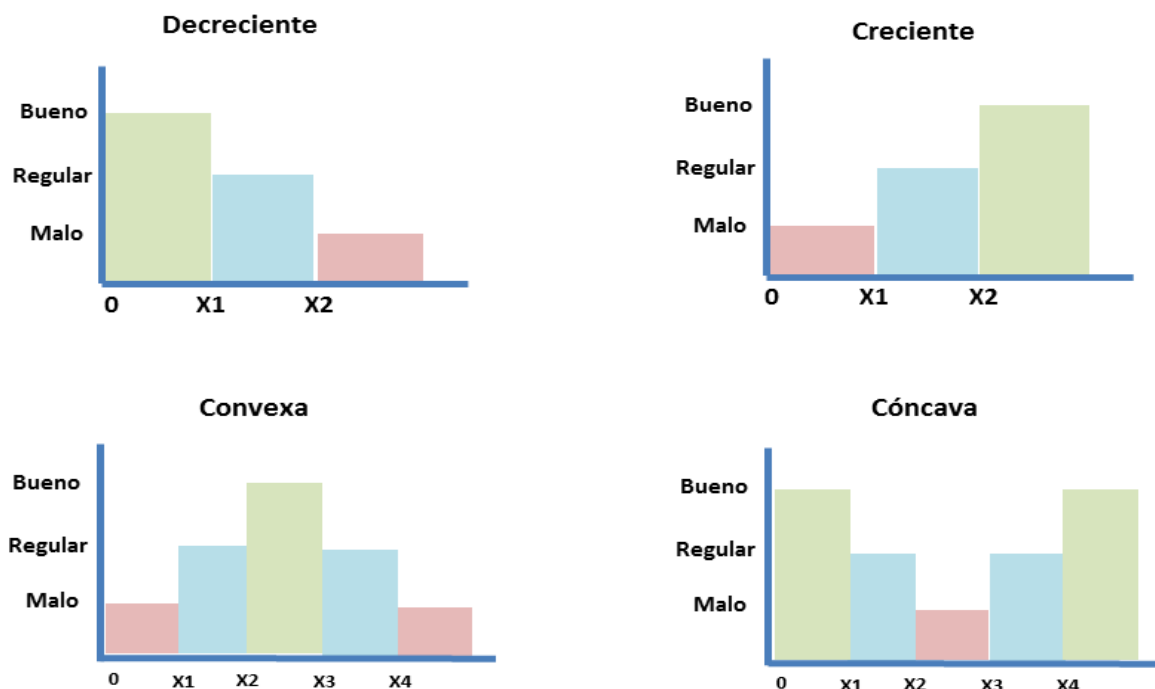


Figura 14 Funciones de paso de un indicador respecto a un factor de un tipo de activo

La clase **Rango**. Es una clase intermedia entre las clases **Factor**, **Indicador** y **TipoActivo** y, además, se relaciona con la clase **EscalaGral** y la clase **FunciónDePaso**. La clase **Rango** representa un intervalo de valores del sistema de medida particular de un indicador respecto a un factor de un tipo de activo. Asimismo, este rango, en base a una función de paso entre el indicador y factor de un tipo de activo, le corresponde una escala general específica. Tener presente que un indicador de un factor de un tipo activo, se relaciona con tres rangos; donde cada rango se corresponde con cada una de las tres escalas generales definidas para nuestro modelo. Para aclarar este punto, nos referimos al ejemplo ya citado en la sección de la escala general. En este ejemplo, descrito en la Tabla 48, referimos a uno de los rangos, específicamente el “De 0 a 1”, que representa al indicador “cantidad de términos imprecisos (CTI)” respecto al factor “inambigüedad” del tipo de activo “requisito individual”. Asimismo, el rango se corresponde con la escala general “bueno” en base a una función de paso “decreciente” según el ejemplo. Los atributos de la clase rango son los siguientes:

- **ValorIni.** Representa el valor inicial del rango concreto.
- **ValorFin.** Representa el valor final del rango concreto.

### 3.3.6 Descripción de las relaciones del modelo de calidad

Después de describir los elementos del modelo de calidad, mostramos ahora las relaciones pertinentes del modelo referentes al proceso de calidad descrito en la presente tesis doctoral. Tener presente que estas relaciones concretas, si bien es cierto son subjetivas, pero tienen su base en la bibliografía revisada y en análisis de las propiedades de los elementos del modelo.



### 3.3.6.1 Relaciones entre “factores” e “indicadores” por tipo de activo

Cada tipo de activo de MORORE tiene características propias que los distinguen uno del otro. En este sentido, para evaluar la calidad de cada tipo de activo, seleccionamos un conjunto de factores con sus respectivos indicadores medibles para cada uno de ellos. Hay que resaltar que un factor puede evaluar a más de un tipo de activo, pero no necesariamente los indicadores son los mismos en esos tipos de activos. Por ejemplo, tenemos, que para evaluar la “atomicidad” de los “requisitos individuales” utilizamos el indicador “número de solapamientos”, mas no utilizamos este indicador para los “patrones de requisitos”. A continuación, mostramos la tabla que relacionan los factores e indicadores por tipo activo. Para la indicación correspondiente utilizamos los siguientes caracteres:

- “R” para los requisitos individuales.
- “P” para los patrones de requisitos.
- “E” para los patrones de estructuras de tipos.

De este modo, si la celda indica “RPE”, quiere decir, que el factor con el indicador correspondiente afecta a los tres tipos de activos. La Tabla 49 muestra las relaciones entre factores e indicadores que permiten evaluar la calidad de los tres tipos de activos de MORORE.

# SOLUCIÓN PROPUESTA

		Atomicidad	Precisión	Abstracción	Inambigüedad	Comprensibilidad	Correctitud	Validabilidad	Verificabilidad	Modificabilidad	Integridad	Frecuencia	Organización	Sencillez	Coherencia	Consistencia	Simplicidad	Adaptabilidad
Morfológicos	Número de elementos de la solución	P								PE	P		E				PE	PE
	Número de palabras.	RP									RP							RP
	Promedio de frases por párrafo.	RP				RPE												RPE
	Promedio de sílabas por palabra.					E												
	Promedio de palabras por oración.					RP								RP				
	Número de abreviaturas o acrónimos.				E	RPE								RP				
	Promedio de signos de puntuación por frase.				RPE	RPE												
Léxicos	Número de términos conectivos.	RP	RP		RPE	RPE					RP							
	Número de términos imprecisos.		RP		RPE				RP									
	Número de términos de diseño.			RP						RP				RP				RP
Analíticos	Número de errores de ortografía y gramática.				RPE	RPE	RP							RP				
	Número de términos ambiguos.		RP		RPE	RPE												
	Número de forma verbales imperativas e indicativas.	R	R															
	Número de formas verbales condicionales y compuestas.		RP											RP				
	Número de términos de dominio.		RP	RP										RP				RPE
Relacionales	Número de relaciones con otros elementos.	RP								RPE								RPE
	Número de dependencias entre elementos del activo.		E							PE			E			PE	PE	PE
	Número de niveles de anidamiento.					E				E			E				E	E
	Número se solapamientos.	R								R								R
Desarrollo	Número de versiones.							R			RP							
	Número de éxitos.											RPE			PE			

Tabla 49 Relación de factores e indicadores que permiten evaluar la calidad de los activos

## SOLUCIÓN PROPUESTA

### 3.3.6.2 Pesos relativos de los indicadores de un factor por activo

Como especificamos anteriormente, los indicadores permiten tener una medida objetiva de un factor respecto a un tipo de activo. Pero no todos los indicadores afectan por igual al factor de un tipo de activo. Para esta diferencia de influencia de los indicadores definimos los pesos relativos de cada indicador respecto a un factor determinado de un tipo de activo. Por ejemplo, para el factor “abstracción” de los “requisitos individuales”, definimos los indicadores: “Número de términos para el diseño (NTDi)” y “Número de términos de dominio (NTDo)”. Considerando que el NTDi tiene el doble de influencia que el NTDo, le asignamos peso 2 al NTDi y peso 1 al NTDo. Recordamos nuevamente que, los valores específicos de los pesos, en este caso, no tienen una gran evidencia o apoyo empírico concreto, pero tienen su base en la bibliografía revisada y en análisis de las propiedades de los elementos del modelo.

#### 3.3.6.2.1 Pesos relativos de los indicadores para requisitos individuales

La Tabla 50 especifica los pesos de los indicadores relativos al factor “Atomicidad” del requisito individual.

	Peso	Porcentaje
Número de palabras	1	0.07
Promedio de frases por párrafo	1	0.07
Número de términos conectivos	4	0.29
Número de forma verbales imperativas e indicativas	3	0.21
Número de relaciones con otros elementos	2	0.14
Número de solapamientos	3	0.21
	14	1

Tabla 50 Pesos de los indicadores para el factor "Atomicidad" del requisito

La Tabla 51 especifica los pesos de los indicadores relativos al factor “Precisión” del requisito individual.

	Peso	Porcentaje
Número de términos conectivos	2	0.15
Número de términos imprecisos	4	0.31
Número de términos ambiguos	3	0.23
Número de forma verbales imperativas e indicativas	2	0.15
Número de formas verbales y condicionales	1	0.08
Número de términos de dominio	1	0.08
	13	1

Tabla 51 Pesos de los indicadores para el factor "Precisión" del requisito

## SOLUCIÓN PROPUESTA

La Tabla 52 especifica los pesos de los indicadores relativos al factor “Abstracción” del requisito individual.

	Peso	Porcentaje
Número de términos de diseño	2	0.67
Número de términos de dominio	1	0.33
	3	1

*Tabla 52 Pesos de los indicadores para el factor "Abstracción" del requisito*

La Tabla 53 especifica los pesos de los indicadores relativos al factor “Inambigüedad” del requisito individual.

	Peso	Porcentaje
Promedio de signos de puntuación por frase	3	0.19
Número de términos conectivos	3	0.19
Número de términos imprecisos	2	0.13
Número de errores de ortografía y gramática.	4	0.25
Número de términos ambiguos	4	0.25
	16	1

*Tabla 53 Pesos de los indicadores para el factor "Inambigüedad" del requisito*

La Tabla 54 especifica los pesos de los indicadores relativos al factor “Comprensibilidad” del requisito individual.

	Peso	Porcentaje
Promedio de frases por párrafo	3	0.17
Promedio de palabras por oración	2	0.11
Número de abreviaturas o acrónimos	2	0.11
Promedio de signos de puntuación por frase	4	0.22
Número de términos conectivos	1	0.06
Número de errores de ortografía y gramática.	3	0.17
Número de términos ambiguos	3	0.17
	18	1

*Tabla 54 Pesos de los indicadores para factor "Comprensibilidad" del requisito*

La Tabla 55 especifica los pesos de los indicadores relativos al factor “Modificabilidad” del requisito individual.

	Peso	Porcentaje
Número de términos de diseño	1	0.20
Número de relaciones con otros elementos.	2	0.40
Número de solapamientos	2	0.40
	5	1

*Tabla 55 Pesos de los indicadores para el factor "Modificabilidad" del requisito*

## SOLUCIÓN PROPUESTA

La Tabla 56 especifica los pesos de los indicadores relativos al factor “Integridad” del requisito individual.

	Peso	Porcentaje
Número de palabras	1	0.33
Número de términos conectivos	1	0.33
Número de versiones.	1	0.33
	3	1

*Tabla 56 Pesos de los indicadores para el factor "Integridad" del requisito*

La Tabla 57 especifica los pesos de los indicadores relativos al factor “Sencillez” del requisito individual.

	Peso	Porcentaje
Promedio de palabras por oración	2	0.18
Número de abreviaturas o acrónimos	1	0.09
Número de términos de diseño	2	0.18
Número de errores de ortografía y gramática.	1	0.09
Número de formas verbales condicionales y compuestas.	2	0.18
Número de términos de dominio	3	0.27
	11	1

*Tabla 57 Pesos de los indicadores para el factor "Sencillez" del requisito*

La Tabla 58 especifica los pesos de los indicadores relativos al factor “Adaptabilidad” del requisito individual.

	Peso	Porcentaje
Número de palabras	3	0.19
Promedio de frases por párrafo	2	0.13
Número de términos de diseño	2	0.13
Número de términos de dominio	3	0.19
Número de relaciones con otros elementos.	4	0.25
Número de solapamientos	2	0.13
	16	1

*Tabla 58 Pesos de los indicadores para el factor "Adaptabilidad" del requisito*

### **3.3.6.2.2 Pesos relativos de los indicadores para patrones de requisitos**

La Tabla 59 especifica los pesos de los indicadores relativos al factor “Atomicidad” del patrón de requisitos.

## SOLUCIÓN PROPUESTA

	Peso	Porcentaje
Número de elementos de la solución	2	0.17
Número de palabras	1	0.08
Promedio de frases por párrafo	1	0.08
Número de términos conectivos	2	0.17
Número de forma verbales imperativas e indicativas	3	0.25
Número de relaciones con otros elementos	3	0.25
	12	1

*Tabla 59 Pesos de los indicadores para el factor "Atomicidad" del patrón de requisitos*

La Tabla 60 especifica los pesos de los indicadores relativos al factor “Precisión” del patrón de requisitos.

	Peso	Porcentaje
Número de términos conectivos	2	0.14
Número de términos imprecisos	4	0.29
Número de términos ambiguos	3	0.21
Número de forma verbales imperativas e indicativas	3	0.21
Número de formas verbales y condicionales	1	0.07
Número de términos de dominio	1	0.07
	14	1

*Tabla 60 Pesos de los indicadores para el factor "Precisión" del patrón de requisitos*

La Tabla 61 especifica los pesos de los indicadores relativos al factor “Abstracción” del patrón de requisitos.

	Peso	Porcentaje
Número de términos de diseño	2	0.67
Número de términos de dominio	1	0.33
	3	1

*Tabla 61 Pesos de los indicadores para el factor "Abstracción" del patrón de requisitos*

La Tabla 62 especifica los pesos de los indicadores relativos al factor “Inambigüedad” del patrón de requisitos.

	Peso	Porcentaje
Promedio de signos de puntuación por frase	3	0.20
Número de términos conectivos	2	0.13
Número de términos imprecisos	2	0.13
Número de errores de ortografía y gramática.	4	0.27
Número de términos ambiguos	4	0.27
	15	1

*Tabla 62 Pesos de los indicadores para el factor "Inambigüedad" del patrón de requisitos*

La Tabla 63 especifica los pesos de los indicadores relativos al factor “Comprensibilidad” del patrón de requisitos.

## SOLUCIÓN PROPUESTA

	Peso	Porcentaje
Promedio de palabras por oración	3	0.18
Promedio de frases por párrafo	1	0.06
Número de abreviaturas o acrónimos	3	0.18
Promedio de signos de puntuación por frase	4	0.24
Número de términos conectivos	1	0.06
Número de errores de ortografía y gramática.	3	0.18
Número de términos ambiguos	2	0.12
	17	1

*Tabla 63 Pesos de los indicadores para el factor "Comprensibilidad" del patrón de requisitos*

La Tabla 64 especifica los pesos de los indicadores relativos al factor “Modificabilidad” del patrón de requisitos.

	Peso	Porcentaje
Número de elementos de la solución	1	0.17
Número de términos de diseño	1	0.17
Número de relaciones con otros elementos.	2	0.33
Número de dependencias entre elementos del activo	2	0.33
	6	1

*Tabla 64 Pesos de los indicadores para el factor "Modificabilidad" del patrón de requisitos*

La Tabla 65 especifica los pesos de los indicadores relativos al factor “Integridad” del patrón de requisitos.

	Peso	Porcentaje
Número de elementos de la solución	2	0.33
Número de palabras	1	0.17
Número de términos conectivos	1	0.17
Número de versiones.	2	0.33
	6	1

*Tabla 65 Pesos de los indicadores para el factor "Integridad" del patrón de requisitos*

La Tabla 66 especifica los pesos de los indicadores relativos al factor “Sencillez” del patrón de requisitos.

	Peso	Porcentaje
Promedio de palabras por oración	2	0.18
Número de abreviaturas o acrónimos	2	0.18
Número de términos de diseño	2	0.18
Número de errores de ortografía y gramática.	1	0.09
Número de formas verbales condicionales y compuestas.	1	0.09
Número de términos de dominio	3	0.27
	11	1

*Tabla 66 Pesos de los indicadores para el factor "Sencillez" del patrón de requisitos*

## SOLUCIÓN PROPUESTA

La Tabla 67 especifica los pesos de los indicadores relativos al factor “Simplicidad” del patrón de requisitos.

	Peso	Porcentaje
Número de elementos de la solución	2	0.50
Número de dependencias entre elementos del activo	2	0.50
	4	1

*Tabla 67 Pesos de los indicadores para el factor "Simplicidad" del patrón de requisitos*

La Tabla 68 especifica los pesos de los indicadores relativos al factor “Adaptabilidad” del patrón de requisitos.

	Peso	Porcentaje
Número de elementos de la solución	3	0.17
Número de palabras	2	0.11
Promedio de frases por párrafo	2	0.11
Número de términos de diseño	2	0.11
Número de términos de dominio	3	0.17
Número de relaciones con otros elementos.	4	0.22
Número de dependencias entre elementos del activo	2	0.11
	18	1

*Tabla 68 Pesos de los indicadores para el factor "Adaptabilidad" del patrón de requisitos*

### **3.3.6.2.3 Pesos relativos de los indicadores para patrones de estructuras de tipos**

La Tabla 69 especifica los pesos de los indicadores relativos al factor “Inambigüedad” del patrón de estructuras de tipos.

	Peso	Porcentaje
Número de abreviaturas y acrónimos	3	0.21
Promedio de signos de puntuación por frase	3	0.21
Número de términos conectivos	1	0.07
Número de términos imprecisos	1	0.07
Número de errores de ortografía y gramática.	3	0.21
Número de términos ambiguos	3	0.21
	14	1

*Tabla 69 Pesos de los indicadores para el factor "Inambigüedad" del patrón de estructuras*

La Tabla 70 especifica los pesos de los indicadores relativos al factor “Comprensibilidad” del patrón de estructuras de tipos.



## SOLUCIÓN PROPUESTA

	Peso	Porcentaje
Promedio de frases por párrafo	1	0.05
Promedio de sílabas por palabra	1	0.05
Número de abreviaturas o acrónimos	3	0.14
Promedio de signos de puntuación por frase	3	0.14
Número de términos conectivos	1	0.05
Número de errores de ortografía y gramática.	3	0.14
Número de términos ambiguos	2	0.10
Número de dependencias entre elementos del activo	4	0.19
Número de niveles de anidamiento	3	0.14
	21	1

*Tabla 70 Pesos de los indicadores para el factor "Comprensibilidad" del patrón de estructuras*

La Tabla 71 especifica los pesos de los indicadores relativos al factor “Modificabilidad” del patrón de estructuras de tipos.

	Peso	Porcentaje
Número de elementos de la solución	3	0.27
Número de relaciones con otros elementos.	2	0.18
Número de dependencias entre elementos del activo	3	0.27
Número de niveles de anidamiento	3	0.27
	11	1

*Tabla 71 Pesos de los indicadores para el factor "Modificabilidad" del patrón de estructuras*

La Tabla 72 especifica los pesos de los indicadores relativos al factor “Organización” del patrón de estructuras de tipos.

	Peso	Porcentaje
Número de dependencias entre elementos del activo	3	0.60
Número de niveles de anidamiento	2	0.40
	5	1

*Tabla 72 Pesos de los indicadores para el factor "Organización" del patrón de estructuras*

La Tabla 73 especifica los pesos de los indicadores relativos al factor “Simplicidad” del patrón de estructuras de tipos.

	Peso	Porcentaje
Número de elementos de la solución	3	0.38
Número de dependencias entre elementos del activo	3	0.38
Número de niveles de anidamiento	2	0.25
	8	1

*Tabla 73 Pesos de los indicadores para el factor "Simplicidad" del patrón de estructuras*

La Tabla 74 especifica los pesos de los indicadores relativos al factor “Consistencia” del patrón de estructuras de tipos.

## SOLUCIÓN PROPUESTA

	Peso	Porcentaje
Número de elementos de la solución	2	0.40
Número de dependencias entre elementos del activo	3	0.60
	5	1

*Tabla 74 Pesos de los indicadores para el factor "Consistencia" del patrón de estructuras*

La Tabla 75 especifica los pesos de los indicadores relativos al factor “Adaptabilidad” del patrón de estructuras de tipos.

	Peso	Porcentaje
Número de elementos de la solución	4	0.25
Promedio de frases por párrafo	1	0.06
Número de términos de dominio	2	0.13
Número de relaciones con otros elementos.	2	0.13
Número de dependencias entre elementos del activo	3	0.19
Número de niveles de anidamiento	4	0.25
	16	1

*Tabla 75 Pesos de los indicadores para el factor "Adaptabilidad" del patrón de estructuras*

### 3.3.6.3 Rangos de valores de indicador-factor por tipo de activo

En las tablas de la sección anterior describimos el peso relativo de los indicadores respecto a un factor de un tipo de activo. Ahora, cuando se realiza una medición de calidad según MORORE, tomamos medidas de los indicadores respectivos de un factor de un tipo de activo. Estas medidas, dependiendo del rango predefinido al que pertenezcan, aportan información que permite evaluar la calidad de un activo. Estos rangos de valores fueron definidos en base a la función de paso (“creciente”, “decreciente”, “cóncava” o “convexa”) del indicador respecto a un factor del tipo de activo a evaluar. Así mismo, el rango indica la escala general a la que pertenecerá el valor particular de la medida del activo. Por ejemplo, tenemos que el rango de valores de 0 – 1 (de 0 a 1) del indicador “Número de términos ambiguos”, del factor “Comprensibilidad” que tiene un tipo de relación “Decrecente”; tiene una escala general de Bueno (1) respecto al tipo de activo requisito individual. Mostramos este ejemplo en la Tabla 86.

Para la especificación de los rangos de MORORE usamos la notación descrita en la Tabla 76. Está notación permitió especificar todos los rangos de valores enteros y decimales descritos para MORORE.

## SOLUCIÓN PROPUESTA

Rango de valores enteros			
Notación	Descripción	Ejemplo notación	Resultado
X-Y	Todos los valores enteros desde X hasta Y, incluidos ambos.	3-5	Valores: 3, 4 y 5
X-	Todos los valores enteros mayores a X, incluido X.	5-	Valores: 5, 6, 7...
-Y	Todos los valores enteros desde cero hasta Y, incluido Y.	-5	Valores: ...3, 4 y 5.
Rangos de valores decimales			
Notación	Descripción	Ejemplo notación	Resultado
]X,Y[	Todos los valores decimales entre X y Y, sin incluir a ambos.	]3,5[	Valores decimales mayores de 3 y menores de 5. Por ejemplo: 3.1; 4; 4.99;
[X,Y]	Todos los valores decimales desde X hasta Y, incluido a ambos.	[3,5]	Valores decimales desde 3 hasta 5. Por ejemplo: 3.0; 3.1; 4.0; 4.99; 5.0
[X,Y[	Todos los valores decimales desde X hasta antes de Y, incluido X pero no Y.	[3,5[	Valores decimales desde 3 hasta antes de 5. Por ejemplo: 3.0; 3.1; 4.0; 4.99.
[X-	Todos los valores decimales mayores a X, incluido X.	[5-	Valores decimales mayores a 5, incluido 5. Por ejemplo: 5; 5.01; 6.3...

*Tabla 76 Notación para la especificación de los rangos de valores*

Asimismo, al igual que las relaciones indicador-factor y los pesos relativos, los valores de los rangos tienen su base en la bibliografía revisada y en el análisis de los elementos del modelo.

### **3.3.6.3.1 Rangos de valores de indicador-factor del requisito individual**

La Tabla 77 especifica los rangos de valores del indicador “Número de palabras” por cada factor al que mide dentro del activo “Requisito individual”.

Número de palabras				
Atomicidad	Integridad	Adaptabilidad	Escala General	
Convexa	Creciente	Decreciente		
50-80	100-	1-50	Bueno	1
21-49 y 81-100	61-100	51-90	Regular	2
1-20 y 100-	1-60	90-	Malo	3

*Tabla 77 Rangos del indicador "número de palabras" por factor del requisito*

La Tabla 78 especifica los rangos de valores del indicador “Promedio de frases por párrafo” por cada factor al que mide dentro del activo “Requisito individual”.

## SOLUCIÓN PROPUESTA

Promedio de frases por párrafo				
Atomicidad	Comprensibilidad	Adaptabilidad	Escala General	
Convexa	Convexa	Decreciente		
[3-4]	[3-4]	[1-4[	Bueno	1
]1-3[ y ]4-5]	]1-3[ y ]4-5]	[4-5[	Regular	2
[1-1] y 5[-	[1-1] y 5[-	[5-	Malo	3

Tabla 78 Rangos del indicador "Promedio de frases por párrafo" por factor del requisito

La Tabla 79 especifica los rangos de valores del indicador “Promedio de palabras por oración” por cada factor al que mide dentro del activo “Requisito individual”.

Promedio de palabras por oración			
Comprensibilidad	Sencillez	Escala General	
Convexa	Decreciente		
[5-12]	[3-8]	Bueno	1
]3-5[ y ]12-18]	]8-12[	Regular	2
[1-3[ y 18[-	12[	Malo	3

Tabla 79 Rangos del indicador "Promedio de palabras por oración" por factor del requisito

La Tabla 80 especifica los rangos de valores del indicador “Número de abreviaturas y acrónimos” por cada factor al que mide dentro del activo “Requisito individual”.

Número de abreviaturas y acrónimos			
Comprensibilidad	Inambigüedad	Escala General	
Decreciente	Decreciente		
0-2	0-2	Bueno	1
3-3	3-3	Regular	2
4-	4-	Malo	3

Tabla 80 Rangos del indicador "Número de abreviaturas y acrónimos" por factor del requisito

La Tabla 81 especifica los rangos de valores del indicador “Promedio de signos de puntuación por frase” por cada factor al que mide dentro del activo “Requisito individual”.

Promedio de signos de puntuación por frase			
Inambigüedad	Comprensibilidad	Escala General	
Convexa	Convexa		
[2-3]	[2-3]	Bueno	1
]3-4]	]3-4]	Regular	2
[0-2[ y 4[-	[0-2[ y 4[-	Malo	3

Tabla 81 Rangos del indicador "Promedio de signos de puntuación por frase" por factor del requisito

La Tabla 82 especifica los rangos de valores del indicador “Número de términos conectivos” por cada factor al que mide dentro del activo “Requisito individual”.

## SOLUCIÓN PROPUESTA

Número de términos conectivos						
Atomicidad	Precisión	Inambigüedad	Comprensibilidad	Integridad	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente	Creciente		
0-2	0-2	0-2	0-2	4-	Bueno	1
3-4	3-4	3-4	3-4	2-3	Regular	2
5-	5-	5-	5-	0-1	Malo	3

*Tabla 82 Rangos del indicador "Número de términos conectivos" por factor del requisito*

La Tabla 83 especifica los rangos de valores del indicador “Número de términos imprecisos” por cada factor al que mide dentro del activo “Requisito individual”.

Número términos imprecisos				
Precisión	Inambigüedad	Verificabilidad	Escala General	
Decreciente	Decreciente	Decreciente		
0-1	0-1	0-1	Bueno	1
2-2	2-2	2-2	Regular	2
3-	3-	3-	Malo	3

*Tabla 83 Rangos del indicador "Número de términos imprecisos" por factor del requisito*

La Tabla 84 especifica los rangos de valores del indicador “Número de términos de diseño” por cada factor al que mide dentro del activo “Requisito individual”.

Número de términos de diseño					
Abstracción	Modificabilidad	Adaptabilidad	Sencillez	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente		
0-1	0-1	0-1	0-1	Bueno	1
2-2	2-2	2-2	2-2	Regular	2
3-	3-	3-	3-	Malo	3

*Tabla 84 Rangos del indicador "Número de términos de diseño" por factor del requisito*

La Tabla 85 especifica los rangos de valores del indicador “Número de errores ortográficos y gramaticales” por cada factor al que mide dentro del activo “Requisito individual”.

Número de errores ortográficos y gramaticales					
Inambigüedad	Comprensibilidad	Correctitud	Sencillez	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente		
0-1	0-1	0-1	0-1	Bueno	1
2-2	2-3	2-2	2-3	Regular	2
3-	4-	3-	4-	Malo	3

*Tabla 85 Rangos del indicador "Número de errores ortográficos y gramaticales" por factor del requisito*

La Tabla 86 especifica los rangos de valores del indicador “Número de términos ambiguos” por cada factor al que mide dentro del activo “Requisito individual”.

## SOLUCIÓN PROPUESTA

Número términos ambiguos				
Precisión	Inambigüedad	Comprensibilidad	Escala General	
Decreciente	Decreciente	Decreciente		
0-1	0-1	0-1	Bueno	1
2-2	2-2	2-2	Regular	2
3-	3-	3-	Malo	3

Tabla 86 Rangos del indicador "Número de términos ambiguos" por factor del requisito

La Tabla 87 especifica los rangos de valores del indicador “Número de formas verbales imperativas e indicativas” por cada factor al que mide dentro del activo “Requisito individual”.

Número de formas verbales imperativas e indicativas			
Atomicidad	Precisión	Escala General	
Convexa	Convexa		
2-3	2-3	Bueno	1
4-4	4-4	Regular	2
0-1 y 5-	0-1 y 5-	Malo	3

Tabla 87 Rangos del indicador "Número de formas verbales imperativas e indicativas" por factor del requisito

La Tabla 88 especifica los rangos de valores del indicador “Número de formas verbales condicionales y compuestas” por cada factor al que mide dentro del activo “Requisito individual”.

Número de formas verbales condicionales y compuestas			
Precisión	Sencillez	Escala General	
Decreciente	Decreciente		
0-2	0-1	Bueno	1
3-3	2-2	Regular	2
4-	3-	Malo	3

Tabla 88 Rangos del indicador "Número de formas verbales condicionales y compuestas" por factor del requisito

La Tabla 89 especifica los rangos de valores del indicador “Número de términos de dominio” por cada factor al que mide dentro del activo “Requisito individual”.

Número de términos de dominio					
Precisión	Abstracción	Sencillez	Adaptabilidad	Escala General	
Convexa	Convexa	Decreciente	Decreciente		
3-3	3-3	0-2	0-2	Bueno	1
2-2	2-2	3-3	3-3	Regular	2
0-1 y 4-	0-1 y 4-	4-	4-	Malo	3

Tabla 89 Rangos del indicador "Número de términos de dominio" por factor del requisito

La Tabla 90 especifica los rangos de valores del indicador “Número de relaciones con otros elementos” por cada factor al que mide dentro del activo “Requisito individual”.

## SOLUCIÓN PROPUESTA

Número de relaciones con otros elementos				
Atomicidad	Modificabilidad	Adaptabilidad	Escala General	
Convexa	Decreciente	Decreciente		
2-3	0-3	0-3	Bueno	1
4-4	4-4	4-4	Regular	2
0-1 y 5-	5-	5-	Malo	3

Tabla 90 Rangos del indicador "Número de relaciones con otros elementos" por factor del requisito

La Tabla 91 especifica los rangos de valores del indicador "Número de solapamientos" por cada factor al que mide dentro del activo "Requisito individual".

Número de solapamientos					
Atomicidad	Comprensibilidad	Modificabilidad	Adaptabilidad	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente		
0-1	0-1	0-1	0-1	Bueno	1
2-2	2-2	2-2	2-2	Regular	2
3-	3-	3-	3-	Malo	3

Tabla 91 Rangos del indicador "Número de solapamientos" por factor del requisito

La Tabla 92 especifica los rangos de valores del indicador "Número de versiones" por cada factor al que mide dentro del activo "Requisito individual".

Número de versiones			
Validabilidad	Integridad	Escala General	
Decreciente	Creciente		
1-2	3-	Bueno	1
3-4	2-2	Regular	2
5-	1-1	Malo	3

Tabla 92 Rangos del indicador "Número de versiones" por factor del requisito

La Tabla 93 especifica los rangos de valores del indicador "Número de éxitos" por cada factor al que mide dentro del activo "Requisito individual".

Número de éxitos		
Frecuencia	Escala General	
Creciente		
4-	Bueno	1
2-3	Regular	2
0-1	Malo	3

Tabla 93 Rangos del indicador "Número de éxitos" por factor del requisito

### 3.3.6.3.2 Rangos de valores de indicador-factor del patrón de requisitos

La Tabla 94 especifica los rangos de valores del indicador "Número de palabras" por cada factor al que mide dentro del activo "Patrón de requisitos".

## SOLUCIÓN PROPUESTA

Número de palabras				
Atomicidad	Integridad	Adaptabilidad	Escala General	
Convexa	Creciente	Decreciente		
351-500	551-	1-350	Bueno	1
151-350 y 501-650	251-550	351-500	Regular	2
1-150 y 651-	1-250	501-	Malo	3

*Tabla 94 Rangos del indicador "Número de palabras" por factor del patrón de requisitos*

La Tabla 95 especifica los rangos de valores del indicador “Número de elementos de la solución” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de elementos de la solución						
Atomicidad	Integridad	Simplicidad	Modificabilidad	Adaptabilidad	Escala General	
Convexa	Creciente	Decreciente	Decreciente	Decreciente		
8-12	10-	3-5	3-5	3-5	Bueno	1
4-7 y 12-15	5-9	6-10	6-10	6-10	Regular	2
-3 y 16-	-4	11-	11-	11-	Malo	3

*Tabla 95 Rangos del indicador "Número de elementos de la solución" por factor del patrón de requisitos*

La Tabla 96 especifica los rangos de valores del indicador “Promedio de frases por párrafo” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Promedio de frases por párrafo				
Atomicidad	Comprensibilidad	Adaptabilidad	Escala General	
Convexa	Convexa	Decreciente		
[3-4]	[3-4]	[1-4[	Bueno	1
]1-3[ y ]4-5]	]1-3[ y ]4-5]	[4-5[	Regular	2
[1-1] y 5[-	[1-1] y 5[-	[5-	Malo	3

*Tabla 96 Rangos del indicador "Promedio de frases por párrafo" por factor del patrón de requisitos*

La Tabla 97 especifica los rangos de valores del indicador “Promedio de palabras por oración” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Promedio de palabras por oración			
Comprensibilidad	Sencillez	Escala General	
Convexa	Decreciente		
[5-12]	[3-8]	Bueno	1
]3-5[ y ]12-18]	]8-12]	Regular	2
[1-3[ y 18[	12[-	Malo	3

*Tabla 97 Rangos del indicador "Promedio de palabras por oración" por factor del patrón de requisitos*

La Tabla 98 especifica los rangos de valores del indicador “Número de abreviaturas y acrónimos” por cada factor al que mide dentro del activo “Patrón de requisitos”.



## SOLUCIÓN PROPUESTA

Número de abreviaturas y acrónimos			
Comprensibilidad	Sencillez	Escala General	
Decreciente	Decreciente		
0-4	0-4	Bueno	1
5-6	5-6	Regular	2
7-	7-	Malo	3

Tabla 98 Rangos del indicador "Número de abreviaturas y acrónimos" por factor del patrón de requisitos

La Tabla 99 especifica los rangos de valores del indicador “Promedio de signos de puntuación por frase” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Promedio de signos de puntuación por frase			
Inambigüedad	Comprensibilidad	Escala General	
Convexa	Convexa		
[2-3]	[2-3]	Bueno	1
]3-4]	[4-4]	Regular	2
[0-2[ y 4[-	[0-2[ y 4[-	Malo	3

Tabla 99 Rangos del indicador "Promedio de signos de puntuación por frase" por factor del patrón de requisitos

La Tabla 100 especifica los rangos de valores del indicador “Número de términos conectivos” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de términos conectivos						
Atomicidad	Precisión	Inambigüedad	Comprensibilidad	Integridad	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente	Creciente		
0-10	0-10	0-8	0-8	13-	Bueno	1
11-15	11-15	9-12	9-12	6-12	Regular	2
16-	16-	13-	13-	0-5	Malo	3

Tabla 100 Rangos del indicador "Número de términos conectivos" por factor del patrón de requisitos

La Tabla 101 especifica los rangos de valores del indicador “Número de términos imprecisos” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número términos imprecisos				
Precisión	Inambigüedad	Verificabilidad	Escala General	
Decreciente	Decreciente	Decreciente		
0-3	0-3	0-3	Bueno	1
4-4	4-4	4-4	Regular	2
5-	5-	5-	Malo	3

Tabla 101 Rangos del indicador "Número de términos imprecisos" por factor del patrón de requisitos

La Tabla 102 especifica los rangos de valores del indicador “Número de términos de diseño” por cada factor al que mide dentro del activo “Patrón de requisitos”.

## SOLUCIÓN PROPUESTA

Número de términos de diseño					
Abstracción	Modificabilidad	Adaptabilidad	Sencillez	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente		
0-2	0-3	0-3	0-3	Bueno	1
3-4	4-4	4-4	4-4	Regular	2
5-	5-	5-	5-	Malo	3

*Tabla 102 Rangos del indicador "Número de términos de diseño" por factor del patrón de requisitos*

La Tabla 103 especifica los rangos de valores del indicador “Número de errores ortográficos y gramaticales” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de errores ortográficos y gramaticales					
Inambigüedad	Comprensibilidad	Correctitud	Sencillez	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente		
0-2	0-2	0-1	0-2	Bueno	1
3-4	3-4	2-3	3-4	Regular	2
5-	5-	4-	5-	Malo	3

*Tabla 103 Rangos del indicador "Número de errores ortográficos y gramaticales" por factor del patrón de requisitos*

La Tabla 104 especifica los rangos de valores del indicador “Número de términos ambiguos” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número términos ambiguos				
Precisión	Inambigüedad	Comprensibilidad	Escala General	
Decreciente	Decreciente	Decreciente		
0-2	0-2	0-2	Bueno	1
3-4	3-3	3-4	Regular	2
5-	4-	5-	Malo	3

*Tabla 104 Rangos del indicador "Número de términos ambiguos" por factor del patrón de requisitos*

La Tabla 105 especifica los rangos de valores del indicador “Número de formas verbales imperativas e indicativas” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de formas verbales imperativas e indicativas			
Atomicidad	Precisión	Escala General	
Convexa	Convexa		
11-15	12-18	Bueno	1
5-10 y 16-20	5-11 y 19-22	Regular	2
0-5 y 21-	0-5 y 22-	Malo	3

*Tabla 105 Rangos del indicador "Número de formas verbales imperativas e indicativas" por factor del patrón de requisitos*

## SOLUCIÓN PROPUESTA

La Tabla 106 especifica los rangos de valores del indicador “Número de formas verbales condicionales y compuestas” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de formas verbales condicionales y compuestas			
Precisión	Sencillez	Escala General	
Decreciente	Decreciente		
0-2	0-2	Bueno	1
3-4	3-4	Regular	2
5-	5-	Malo	3

*Tabla 106 Rangos del indicador "Número de formas verbales condicionales y compuestas" por factor del patrón de requisitos*

La Tabla 107 especifica los rangos de valores del indicador “Número de términos de dominio” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de términos de dominio					
Precisión	Abstracción	Sencillez	Adaptabilidad	Escala General	
Convexa	Convexa	Decreciente	Decreciente		
4-10	4-10	0-8	0-8	Bueno	1
2-3 y 11-12	2-3 y 11-12	9-12	9-12	Regular	2
0-1 y 13-	0-1 y 13-	13-	13-	Malo	3

*Tabla 107 Rangos del indicador "Número de términos de dominio" por factor del patrón de requisitos*

La Tabla 108 especifica los rangos de valores del indicador “Número de relaciones con otros elementos” por cada factor al que mide dentro del activo “Patrón de requisitos”.

Número de relaciones con otros elementos				
Atomicidad	Modificabilidad	Adaptabilidad	Escala General	
Decreciente	Decreciente	Decreciente		
0-1	0-2	0-2	Bueno	1
2-2	3-3	3-3	Regular	2
3-	4-	4-	Malo	3

*Tabla 108 Rangos del indicador "Número de relaciones con otros elementos" por factor del patrón de requisitos*

La Tabla 109 especifica los rangos de valores del indicador “Número de dependencias entre elementos del activo” por cada factor al que mide dentro del activo “Patrón de requisitos”.

## SOLUCIÓN PROPUESTA

Número de dependencias entre elementos del activo					
Consistencia	Simplicidad	Modificabilidad	Adaptabilidad	Escala General	
Cóncavo	Decreciente	Decreciente	Decreciente		
0-1 y 5-	0-3	0-3	0-3	Bueno	1
2-2 y 4-4	4-4	4-4	4-4	Regular	2
3-3	5-	5-	5-	Malo	3

Tabla 109 Rangos del indicador "Número de dependencias entre elementos del activo" por factor del patrón de requisitos

La Tabla 110 especifica los rangos de valores del indicador "Número de versiones" por cada factor al que mide dentro del activo "Patrón de requisitos".

Número de versiones		
Integridad	Escala General	
Creciente		
3-	Bueno	1
2-2	Regular	2
1-1	Malo	3

Tabla 110 Rangos del indicador "Número de versiones" por factor del patrón de requisitos

La Tabla 111 especifica los rangos de valores del indicador "Número de éxitos" por cada factor al que mide dentro del activo "Patrón de requisitos".

Número de éxitos			
Frecuencia	Coherencia	Escala General	
Creciente	Creciente		
4-	3-	Bueno	1
2-3	2-2	Regular	2
0-1	0-1	Malo	3

Tabla 111 Rangos del indicador "Número de éxitos" por factor del patrón de requisitos

### 3.3.6.3.3 Rangos de valores de indicador-factor del activo patrón de estructuras de tipos

La Tabla 112 especifica los rangos de valores del indicador "Número de elementos de la solución" por cada factor al que mide dentro del activo "Patrón de estructuras de tipos".

Número de elementos de la solución						
Modificabilidad	Organización	Simplicidad	Consistencia	Adaptabilidad	Escala General	
Decreciente	Convexo	Decreciente	Convexo	Decreciente		
2-5	4-6	2-5	4-6	2-5	Bueno	1
6-7	3-3 y 7-8	6-7	3-3 y 7-8	6-7	Regular	2
8-	0-2 y 9-	8-	0-2 y 9-	8-	Malo	3

Tabla 112 Rangos del indicador "Número de elementos de la solución" por factor del patrón de estructuras

## SOLUCIÓN PROPUESTA

La Tabla 113 especifica los rangos de valores del indicador “Promedio de frases por párrafo” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Promedio de frases por párrafo			
Comprensibilidad	Adaptabilidad	Escala General	
Convexa	Creciente		
[3-4]	[1-4[	Bueno	1
]1-3[ y ]4-5]	[4-5[	Regular	2
[0-1] y 5[-	[5-	Malo	3

Tabla 113 Rangos del indicador "Promedio de frases por párrafo" por factor del patrón de estructuras

La Tabla 114 especifica los rangos de valores del indicador “Promedio de sílabas por palabra” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Promedio de sílabas por palabras		
Comprensibilidad	Escala General	
Convexa		
[4-5]	Bueno	1
[2-4[ y ]5-6]	Regular	2
[1-2[ y 6[-	Malo	3

Tabla 114 Rangos del indicador "Promedio de sílabas por palabra" por factor del patrón de estructuras

La Tabla 115 especifica los rangos de valores del indicador “Número de abreviaturas y acrónimos” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de abreviaturas y acrónimos			
Comprensibilidad	Inambigüedad	Escala General	
Decreciente	Decreciente		
0-2	0-2	Bueno	1
3-3	3-3	Regular	2
4-	4-	Malo	3

Tabla 115 Rangos del indicador "Número de abreviaturas y acrónimos" por factor del patrón de estructuras

La Tabla 116 especifica los rangos de valores del indicador “Número de signos de puntuación por frase” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de signos de puntuación por frase			
Inambigüedad	Comprensibilidad	Escala General	
Convexa	Convexa		
2-3	2-3	Bueno	1
4-4	4-4	Regular	2
0-1 y 5-	0-1 y 5-	Malo	3

Tabla 116 Rangos del indicador "Número de signos de puntuación por frase" por factor del patrón de estructuras

## SOLUCIÓN PROPUESTA

La Tabla 117 especifica los rangos de valores del indicador “Número de términos conectivos” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de términos conectivos			
Inambigüedad	Comprensibilidad	Escala General	
Decreciente	Decreciente		
0-8	0-8	Bueno	1
9-15	9-15	Regular	2
16-	16-	Malo	3

Tabla 117 Rangos del indicador "Número de términos conectivos" por factor del patrón de estructuras

La Tabla 118 especifica los rangos de valores del indicador “Número de términos imprecisos” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de términos imprecisos		
Inambigüedad	Escala General	
Decreciente		
0-1	Bueno	1
2-3	Regular	2
4-	Malo	3

Tabla 118 Rangos del indicador "Número de términos imprecisos" por factor del patrón de estructuras

La Tabla 119 especifica los rangos de valores del indicador “Número de errores ortográficos y gramaticales” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de errores ortográficos y gramaticales			
Inambigüedad	Comprensibilidad	Escala General	
Decreciente	Decreciente		
0-2	0-2	Bueno	1
3-3	3-3	Regular	2
4-	4-	Malo	3

Tabla 119 Rangos del indicador "Número de errores ortográficos y gramaticales" por factor del patrón de estructuras

La Tabla 120 especifica los rangos de valores del indicador “Número de términos ambiguos” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número términos ambiguos			
Inambigüedad	Comprensibilidad	Escala General	
Decreciente	Decreciente		
0-1	0-1	Bueno	1
2-3	2-3	Regular	2
4-	4-	Malo	3

Tabla 120 Rangos del indicador "Número de términos ambiguos" por factor del patrón de estructuras

## SOLUCIÓN PROPUESTA

La Tabla 121 especifica los rangos de valores del indicador “Número de términos de dominio” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de términos de dominio		
Adaptabilidad	Escala General	
Decreciente		
0-6	Bueno	1
7-10	Regular	2
11-	Malo	3

Tabla 121 Rangos del indicador "Número de términos de dominio" por factor del patrón de estructuras

La Tabla 122 especifica los rangos de valores del indicador “Número de relaciones con otros elementos” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de relaciones con otros elementos			
Modificabilidad	Adaptabilidad	Escala General	
Decreciente	Decreciente		
0-2	0-2	Bueno	1
3-3	3-3	Regular	2
4-	4-	Malo	3

Tabla 122 Rangos del indicador "Número de relaciones con otros elementos" por factor del patrón de estructuras

La Tabla 123 especifica los rangos de valores del indicador “Número de dependencias entre elementos del activo” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de dependencias entre elementos del activo							
Comprensibilidad	Simplicidad	Modificabilidad	Adaptabilidad	Consistencia	Organización	Escala General	
Convexa	Decreciente	Decreciente	Decreciente	Convexa	Convexa		
3-4	0-2	0-2	0-2	3-4	3-4	Bueno	1
2-2 y 5-5	3-4	3-4	3-4	2-2 y 5-5	2-2 y 5-5	Regular	2
0-1 y 6-	5-	5-	5-	0-1 y 6-	0-1 y 6-	Malo	3

Tabla 123 Rangos del indicador "Número de dependencias entre elementos del activo" por factor del patrón de estructuras

La Tabla 124 especifica los rangos de valores del indicador “Número de niveles de anidamiento” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

## SOLUCIÓN PROPUESTA

Número de niveles de anidamiento						
Comprensibilidad	Simplicidad	Modificabilidad	Adaptabilidad	Organización	Escala General	
Decreciente	Decreciente	Decreciente	Decreciente	Convexa		
0-2	0-2	0-2	0-2	2-3	Bueno	1
3-4	3-4	3-4	3-4	4-5	Regular	2
5-	5-	5-	5-	0-1 y 6-	Malo	3

Tabla 124 Rangos del indicador "Número de niveles de anidamiento" por factor del patrón de estructuras

La Tabla 125 especifica los rangos de valores del indicador “Número de éxitos” por cada factor al que mide dentro del activo “Patrón de estructuras de tipos”.

Número de éxitos			
Frecuencia	Coherencia	Escala General	
Creciente	Creciente		
3-	3-	Bueno	1
2-2	2-2	Regular	2
0-1	0-1	Malo	3

Tabla 125 Rangos del indicador "Número de éxitos" por factor del patrón de estructuras

### 3.3.7 EJEMPLO: análisis de un patrón de requisitos

En la sección anterior describimos el modelo orientado a evaluar la calidad de los tipos de activos definidos en MORORE. Especificamos los elementos que intervienen en el sistema de métricas para realizar una medición efectiva. Ahora en esta sección, con ayuda de un ejemplo, mostraremos mayor detalle del proceso para comprender mejor los procedimientos que involucra.

Para evaluar la calidad de un requisito, un patrón de requisitos o un patrón de estructuras, primero debemos identificar cuáles son los factores que se medirán. Estos factores dependerán precisamente del tipo de activo, sin embargo, como se ha mencionado antes, un mismo factor puede afectar a uno o más activos y un activo puede ser afectado por uno o más factores.

En este caso, evaluamos la calidad de un patrón de requisitos, por ejemplo, en la Tabla 126, describimos un patrón de requisitos de “gestión de inventarios” formado por trece requisitos definidos como requisitos funcionales. A primera vista los requisitos no están del todo claros. Por ejemplo, en el primer requisito se habla de agregar, modificar, deshabilitar y eliminar artículos. Pero al utilizar el verbo “podría” no deja claro si el sistema debe o no debe permitir estas funciones o si son requeridas realmente por el cliente. Lo mismo ocurre con el requisito siete, que habla de visualizar datos registrados, pero sólo en ciertas ocasiones, ¿Cuáles ocasiones?, ¿Cuándo no se podrán visualizar? El requisito no deja claro lo que se necesita. Además, en algunos casos, los requisitos son rígidos. Por ejemplo, el requisito habla de los datos de los usuarios y se especifica que cada usuario debe tener sólo los datos indicados, afectando la adaptabilidad de los requisitos. Por otro lado, si nos fijamos en otros atributos como la descripción del patrón, podemos notar qué, si bien es clara, podría expresarse de forma más concisa. De igual



## SOLUCIÓN PROPUESTA

---

modo con el atributo que expresa las ventajas y desventajas, aquí se puede utilizar un lenguaje más claro y proporcionar más información.

En las secciones anteriores se ha definido cuáles son los factores por los que son afectados los patrones de requisitos. Tomamos el modelo en cuenta para realizar el análisis de nuestro patrón. Estos factores son: Atomicidad, Precisión, Abstracción, Inambigüedad, Comprensibilidad, Correctitud, Verificabilidad, Modificabilidad, Integridad, Frecuencia, Sencillez, Coherencia, Consistencia, Simplicidad y Adaptabilidad (Tabla 49).

Después de identificar los factores, en cada uno de ellos se evaluó una serie de indicadores de diversos tipos (morfológicos, léxicos, relacionales, etc.). Estos Indicadores, que permiten la evaluación objetiva de los factores del patrón, están definidos también en la Tabla 49.

Para realizar la evaluación en cuestión, fue necesario tomar en cuenta los elementos siguientes:

- La función de paso (creciente, decreciente, convexa, cóncava). Desde la Tabla 94 hasta la Tabla 111.
- Los rangos de valores. Desde la Tabla 94 hasta la Tabla 111.
- El peso relativo del indicador. Desde la Tabla 59 hasta la Tabla 68.

## SOLUCIÓN PROPUESTA

Identificador:	PR00000001	
Nombre:	Gestión de inventarios.	
Tipo:	Patrón de requisitos.	
Descripción:	El patrón "Gestión de inventarios" controla el ingreso y salida de artículos dentro de un almacén. Además, permite registrar nuevos artículos y eliminar antiguos. También controla a los usuarios que registran algún cambio en el sistema. Finalmente, permite emitir reportes de las actividades registradas en el almacén.	
Ventajas:	Proporciona un manejo y control del inventario básico que se orienta al orden y fácil manejo del mismo. Rápida solución para un almacén con lista de productos que no cambia mucho en el tiempo.	
Desventajas:	No guarda un historial de artículos eliminados, por lo cual si se desea reiniciar el almacenamiento de algún producto, este aparecerá como si fuera uno nuevo.	
Palabras claves:	Gestión de almacén, altas y bajas de productos, control de usuarios.	
Requisitos:	FUNC-00001	Se podría, usualmente, agregar y eliminar artículos; también modificar artículos, y deshabilitar artículos.
	FUNC-00002	Los artículos poseen cinco datos: Nombre, Tipo, Precio, Stock, estado; y el nombre siempre tiene como máximo cien caracteres.
	FUNC-00003	El sistema, por lo general, puede agregar entradas y salidas; así como devoluciones de artículos, según corresponda.
	FUNC-00004	Las entradas poseen siempre los siguientes datos: Usuario, Proveedor, Hora, Fecha, Artículo, Cantidad, Precio
	FUNC-00005	Las salidas poseen siempre los siguientes datos: Usuario, Solicitante, Hora, Fecha, Artículo, Cantidad, Precio
	FUNC-00006	Las devoluciones poseen, cuando no se indique lo contrario, los siguientes datos, que son llenados, por el jefe de almacén: Usuario, Solicitante, Hora, Fecha, Artículo, Cantidad, Precio
	FUNC-00007	A veces se podrá visualizar artículos, entradas, salidas y devoluciones.
	FUNC-00008	Siempre se podrá agregar usuarios. Siempre se podrá modificar usuarios. Siempre se podrá deshabilitar usuarios. Siempre se podrá eliminar usuarios.
	FUNC-00009	Los usuarios, solamente, tendrían los siguientes datos: Nombre, Identificador, Tipo, Estado
	FUNC-00010	Algunas veces, se podrá agregar, modificar y deshabilitar; y también eliminar tipos de usuario.
	FUNC-00011	Los tipos de usuario tienen, solamente, los siguientes datos y son dos: Nombre, Descripción.
	FUNC-00012	Siempre se podrá registrar tipos de artículos
	FUNC-00013	Los tipos de artículos, pueden tener, los siguientes datos, que son solamente dos: Nombre, descripción.

*Tabla 126 Ejemplo de patrón de requisitos mejorable.*

La función de paso entre indicador y factor de un tipo de activo influye en la fijación de los rangos de la escala particular de cada indicador. Cada rango corresponde a un valor de la escala general. Así, por ejemplo, para el Indicador “A” tenemos los rangos -]2, [2,6] y 6[- para valores particulares, que corresponden a los valores cualitativos “Malo”, “Regular” y “Bueno” y los valores cuantitativos 3, 2 y 1 de la escala general respectivamente. En este caso existe una relación creciente entre el factor y el Indicador “A”. Sin embargo, si la relación fuera decreciente, el rango con los valores mayores

## SOLUCIÓN PROPUESTA

correspondería a “Malo” o 3. En general los rangos fueron ubicados de acuerdo al tipo de relación factor–indicador, por lo que fue importante considerarlo antes de realizar la evaluación.

Por su parte, el peso relativo que expresa el hecho de que un indicador prevalezca sobre otro, fue utilizado para obtener la evaluación final del factor. Este proceso lo mostramos más adelante.

Los rangos de valores de cada uno de los indicadores para cada factor por tipo de activo, ya han sido establecidos desde la Tabla 94 a la Tabla 111. Del mismo modo los pesos de los indicadores respecto a cada factor de los patrones de requisitos están desde la Tabla 59 a la Tabla 68. Entonces, en base a estos rangos dimos un valor cualitativo general al indicador a partir del valor particular que se obtuvo al medir el patrón.

Por ejemplo, para un patrón de requisitos, el indicador “número de términos conectivos” (NTC) y el indicador “número de términos ambiguos” (NTA) tienen los rangos que se muestran en la Tabla 127 y en la Tabla 128 para el factor “Comprensibilidad”:

Rango	Escala General	
0–8	Bueno	1
9–12	Regular	2
13–	Malo	3

*Tabla 127 Rango de valores del indicador "número de términos conectivos" respecto al factor "Comprensibilidad" de un patrón de requisitos (obtenido de la Tabla 100)*

Rango	Escala General	
0–2	Bueno	1
3–4	Regular	2
5–	Malo	3

*Tabla 128 Rango de valores del indicador "número de términos ambiguos" respecto al factor "Comprensibilidad" de un patrón de requisitos (obtenido de la Tabla 104)*

Al analizar el patrón (Tabla 126), contamos 20 términos conectivos y 1 término ambiguo. De acuerdo a los rangos de valores expuestos, corresponden a una calificación de “Malo” para el indicador NTC y “Bueno” para el indicador NTA. De igual modo tendríamos que utilizar los rangos establecidos de los mismos indicadores para otro factor al que afecten. Por ejemplo, para el factor “Integridad” el “número de términos conectivos” tiene los rangos que se ven en la Tabla 129. Estos rangos responden a una función de paso “creciente” entre la “integridad” y el NTC para el patrón de requisitos. Entonces, en base a los 20 términos conectivos encontrados en el patrón ejemplo, el NTC califica con “Bueno” a la “Integridad” y al mismo tiempo con “Malo” a la comprensibilidad del patrón.

## SOLUCIÓN PROPUESTA

Rango	Escala General	
13-	Bueno	1
6-12	Regular	2
0-5	Malo	3

*Tabla 129 Rango de valores del indicador "número de términos conectivos" respecto al factor "Integridad" de un patrón de requisitos (obtenido de la Tabla 100)*

Una calificación cualitativa de cada indicador aún no permite apreciar de forma clara si el patrón es adecuado o no, ni qué factores deben mejorarse. Por un lado, el “número de términos conectivos” nos dice que la “comprensibilidad” del patrón es mala, mientras que el “número de términos ambiguos” la califica como buena. Entonces, ¿qué tan bueno es el patrón respecto a su comprensibilidad?

La evaluación final de un factor la hallamos mediante el promedio ponderado de las calificaciones que asignen sus indicadores. Para efectuar dicha ponderación es necesario trabajar con el valor cuantitativo correspondiente de la Escala General. Los valores cuantitativos de la escala general están definidos como 1, 2 y 3 para las calificaciones de “Bueno”, “Regular” y “Malo” respectivamente. En este caso al indicador NTC le correspondería el valor 3 y al indicador NTA el valor 1.

Para hallar la “nota final” (NF) que obtiene el patrón en cada factor evaluado, ponderamos los valores cuantitativos que obtuvieron cada uno de los indicadores con sus pesos respectivos, los que fueron fijados anteriormente (Tabla 63). Las operaciones a realizar están descritas en Fórmula 1. Donde  $Nota_i$  representan el valor cuantitativo en la escala general y  $Peso_i$  el peso de cada indicador, siendo  $n$  el número de indicadores.

$$\frac{\sum_{i=1}^n (Nota_i \cdot Peso_i)}{\sum_{i=1}^n Peso_i}$$

*Fórmula 1 Fórmula para obtener el promedio ponderado de calidad de un factor en base a sus indicadores*

Volviendo al ejemplo, los indicadores NTC y NTA tienen un peso de 1 y 2 respectivamente en el factor “Comprensibilidad” de un patrón de requisitos (Tabla 63). Sí suponemos que estos dos fueran los únicos indicadores que afectan a la comprensibilidad del patrón, entonces para obtener la nota final de la comprensibilidad aplicamos la Fórmula 1. El resultado de la aplicación lo mostramos en la Fórmula 2.

$$NF = \frac{\sum_{i=1}^n (Nota_i * Peso_i)}{\sum_{i=1}^n Peso_i} = \frac{3*1 + 1*2}{1 + 2} = 1.67$$

*Fórmula 2 Aplicación de la fórmula para obtener el promedio ponderado de calidad de un factor en base a sus indicadores*

## SOLUCIÓN PROPUESTA

---

Este resultado indica que el promedio ponderado para la comprensibilidad del patrón está entre 1 y 2, en forma cualitativa, en la escala general, el resultado está entre “Regular” y “Bueno”.

Se realiza esta operación en todos los factores y para así obtener un valor numérico en cada uno de ellos. De esta manera tendremos una visión detallada de que aspectos tendremos que mejorar del activo evaluado.

En la Tabla 130 se muestra los pesos y el valor cuantitativo de la escala general obtenido de cada indicador, además la nota final ponderada en cada factor.

Una vez obtenido los valores de cada factor en la Tabla 130, es necesario interpretarlos. Para esta interpretación se recurre nuevamente a la escala general. Si el valor final es decimal, tendremos que interpretarlo de acuerdo a la proximidad que tenga a los valores de la escala general entre los que se encuentra.

En el caso de este patrón se observa que necesita mejorar en primer lugar en cuanto a los factores “Correctitud” y “Verificabilidad” (valores de nota final en “**negrita**” en la Tabla 130). Los valores obtenidos en estos dos factores corresponden exactamente a la calificación “Malo”, por lo que se deberá darles mayor prioridad. Además, hay algunos factores que obtuvieron una calificación “Regular” o cuyo valor está muy cercano ya sea por algunos decimales más o algunos menos, se puede mencionar aquí a la “Atomicidad”, “Precisión”, “Inambigüedad”, “Comprensibilidad”, “Integridad”, “Frecuencia”, “Sencillez” y “Simplicidad” (valores de nota final en “*cursiva*” en la Tabla 130).

Lo siguiente por hacer es, analizar por qué el patrón necesita revisión respecto a los factores mencionados. Si se mira la calificación de los indicadores, encontramos que existen cuatro indicadores principales calificados en mayor medida como “Malo”, y son precisamente estos cuatro los que afectan a los factores con baja calificación.

Estos indicadores a mejorar son: “número de términos imprecisos”, “número de errores de ortografía y gramática”, “número de términos conectivos” y el “promedio de frases por párrafo” (indicadores y valores en “**negrita**” en la Tabla 130). Si se mejora estos cuatro indicadores, mejorarán los factores observados. Entonces lo siguiente es revisar la redacción de los atributos del patrón y volver a analizarlo.

Para poder hacer una comparación, en la Tabla 131, especificamos al patrón de requisitos que obtendría una mejor calificación. Es decir, presentamos al patrón que mejoramos las deficiencias especificadas por la evaluación realizada.

## SOLUCIÓN PROPUESTA

		Atomicidad		Precisión		Abstracción		Inambigüedad		Comprensibilidad		Correctitud		Verificabilidad		Modificabilidad		Integridad		Frecuencia		Sencillez		Coherencia		Consistencia		Simplicidad		Adaptabilidad	
Morfológicos	Número de elementos de la solución.	0.22	2													0.17	3	0.33	1									0.5	3	0.17	3
	Número de palabras.	0.11	2															0.17	2										0.11	1	
	Promedio de palabras por oración									0.18	2											0.18	3								
	Promedio de frases por párrafo.	0.11	2							0.06	2												1						0.11	1	
	Número de abreviaturas o acrónimos									0.18	1											0.18									
	Número de signos de puntuación por frase.							0.2	1	0.24	1																				
Léxicos	Número de términos conectivos.	0.22	3	0.23	3			0.13	3	0.06	3							0.17	1												
	Número de términos imprecisos.			0.31	3			0.13	3					1	3																
	Número de términos de diseño					0.67	1									0.17	1					0.18	1						0.11	1	

## SOLUCIÓN PROPUESTA

Criterios de Evaluación	Analíticos				Relacionales				Desarrollo																						
	Número de errores de ortografía y gramática.						0.27	3	0.18	3	1	3						0.09	3												
	Número de términos ambiguos.			0.23	1			0.27	1	0.12	1																				
	Número de formas verbales condicionales y compuestas.			0.08	2													0.09	2												
	Número de términos de dominio.			0.15	1	0.33	1											0.27	2							0.17	2				
	Número de relaciones con otros elementos.	0.33	1									0.33	1													0.22	1				
	Número de dependencias entre elementos del activo.											0.33										1		0.5		0.11					
	Número de versiones.												0.33	3																	
	Número de éxitos.													1	2				1	1											
Nota		1.87		2.16		1.00		2.06		1.74		3.00		3.00		1.34		1.83		2.00		1.71		1.00		1.00		2.00		1.51	
		Atomicidad		Precisión		Abstracción		Inambigüedad		Comprensibilidad		Correctitud		Verificabilidad		Modificabilidad		Integridad		Frecuencia		Sencillez		Coherencia		Consistencia		Simplicidad		Adaptabilidad	

*Tabla 130 Resultados del análisis del patrón de requisitos "Gestión de Inventarios"*

## SOLUCIÓN PROPUESTA

Identificador:	PR00000001	
Nombre:	Gestión de inventarios.	
Tipo:	Patrón de requisitos.	
Descripción:	El patrón "Gestión de inventarios" realiza los siguiente: Controla el ingreso y salida de artículos de un almacén. Registra y elimina artículos. Controla a los usuarios que realizan cambios en el sistema. Permite emitir reportes de las actividades del almacén.	
Ventajas:	Controla el inventario básico que se orienta al orden y fácil manejo del mismo. Es una solución para un almacén con lista de productos que no cambia mucho en el tiempo.	
Desventajas:	No guardar un historial de artículos eliminados, reiniciamos el almacenamiento de un producto, este aparecerá como si fuera uno nuevo.	
Palabras claves:	Gestión de almacén, altas y bajas de productos, control de usuarios.	
Requisitos:	FUNC-00001	El software permite al usuario agregar, modificar y eliminar artículos.
	FUNC-00002	Los artículos al menos deben tener al menos los siguientes atributos: Nombre (100 caracteres como máximo), Tipo, Precio, Stock y Estado.
	FUNC-00003	El software permite al usuario registrar entradas, salidas y devoluciones (Especificar el motivo).
	FUNC-00004	Las entradas al menos deben tener los siguientes datos: Usuario, Proveedor, Hora, Fecha, Artículo, Cantidad y Precio.
	FUNC-00005	Las salidas al menos deben tener los siguientes datos: Usuario, Solicitante, Hora, Fecha, Artículo, Cantidad y Precio.
	FUNC-00006	Las devoluciones poseen, cuando no se indique lo contrario, los siguientes datos especificados por el jefe de almacén: Usuario, Solicitante, Hora, Fecha, Artículo, Cantidad, Precio
	FUNC-00007	El software debe permitir al usuario visualizar artículos, entradas, salidas y devoluciones.
	FUNC-00008	El software debe permitir al jefe de almacén agregar, modificar, eliminar y deshabilitar usuarios.
	FUNC-00009	Los usuarios al menos deben tener los siguientes datos: Nombre, Identificador, Tipo y Estado
	FUNC-00010	El software debe permitir al jefe de almacén agregar, modificar, eliminar y deshabilitar.
	FUNC-00011	Los tipos de usuario al menos deben tener los siguientes datos: Nombre y Descripción.
	FUNC-00012	El software debe permitir al usuario registrar tipos de artículos.
	FUNC-00013	Los tipos de artículos al menos deben tener los siguientes datos: Nombre y Descripción.

*Tabla 131 Ejemplo de Patrón de Requisitos Mejorado*

### 3.4 Modelo del proceso de reúso para MORORE

Como complemento a la descripción de MORORE, en esta sección, presentamos el modelo del proceso que permita el reúso efectivo y eficiente de activos propuestos de la ingeniería de requisitos. Tenemos que resaltar que, el contenido de esta sección fue



## SOLUCIÓN PROPUESTA

revisado en un proyecto fin de carrera (PFC) propuesto y dirigido por el autor de la presente tesis doctoral (Cáceres, 2014).

Todo proceso de reuso parte del supuesto de que existe previamente una base de conocimiento: conjunto de elementos que serán los que se van consultar y reutilizar. Esta base de conocimiento evidencia la necesidad de un proceso adicional, el cual garantice la creación y el mantenimiento del repositorio. De esta manera, podemos adelantar que una metodología de reuso debe estar conformada por dos procesos complementarios e interdependientes: guardar y consultar, que vamos a explicar a continuación.

La metodología desarrollada en esta tesis consiste en la combinación de un proceso de preparación de los elementos hasta convertirse en activos de software reutilizables y almacenables; y de un proceso de extracción de estos activos para aplicarlos a una situación particular. En la Figura 15 mostramos el modelo de proceso de reuso de MORORE.

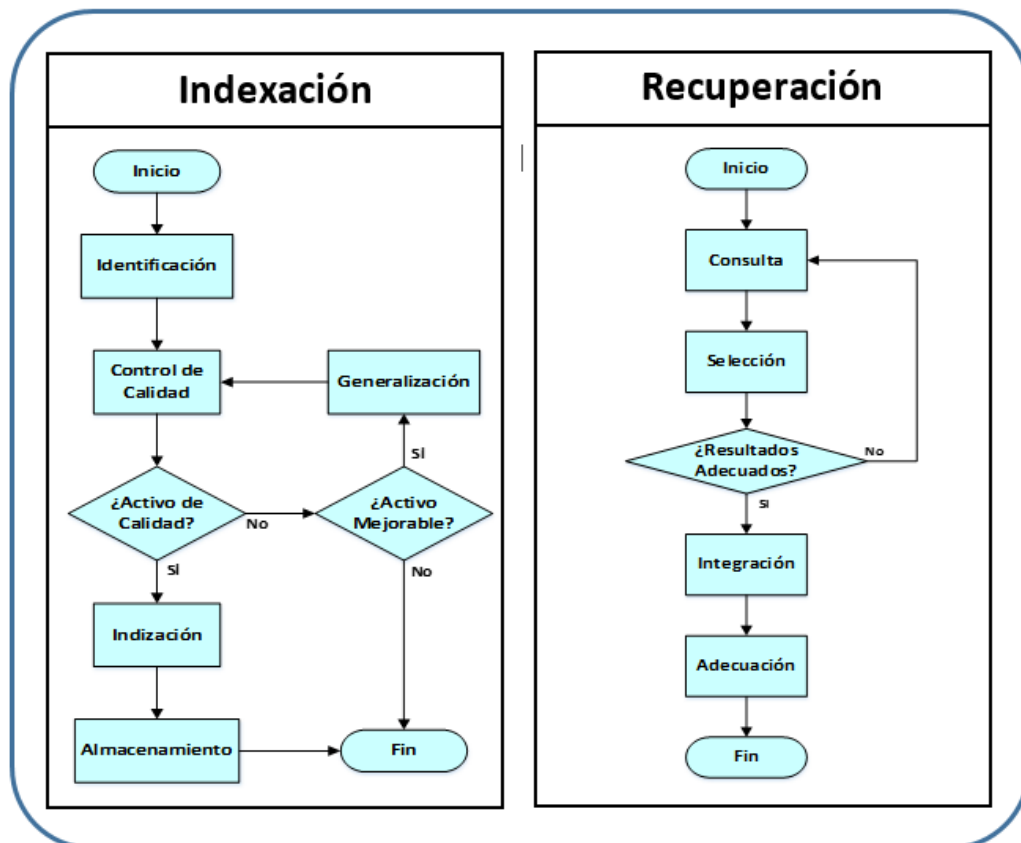


Figura 15. Proceso de indexación y de recuperación propuesto.

En nuestra metodología, hemos denominado *Indexación* al procedimiento que debe seguir el analista para preparar un activo de requisitos, para que pueda ser almacenado en un repositorio. Este nombre *Indexación* hace referencia al objetivo final de este proceso: registrar ordenadamente toda la información del activo para incluirlo en el repositorio. Una adecuada indexación permite consultar activos de forma sustancialmente más rápida y relevante al momento de realizar una búsqueda en el

## SOLUCIÓN PROPUESTA

---

catálogo. Estas mejoras facilitan el reúso en los proyectos de software y lo hacen más atractivo para los analistas.

El proceso de *Indexación*, según observamos en el panel izquierdo de la Figura 15, inicia con la *Identificación* de los activos de requisitos, luego el activo es sometido a un control de calidad. Si se determina que el activo no tiene la calidad suficiente, pero se puede mejorar, es generalizado y luego sometido nuevamente al control de calidad. Si se determina que el activo no tiene la calidad suficiente y no se puede mejorar se descarta terminando el proceso. Por otro lado, si se determina que, el activo está apto para su almacenamiento, se indiza y finalmente es almacenado en un catálogo.

Es relevante mencionar la importancia de incluir constantemente nuevos activos al catálogo. El repositorio no se debe considerar como un producto final estático, sino como un repositorio perfectible en continua evolución. El ingeniero debe ser consciente de que su rol no es simplemente de usuario del repositorio y que sus aportes son fundamentales para mantenerlo actualizado.

Asumiendo que ya se dispone de un repositorio de activos de requisitos almacenados, los ingenieros pueden consultar, seleccionar y reusar los activos de requisitos que más se adapten a sus necesidades para elaborar la especificación del nuevo proyecto en el que están trabajando. Como ya mencionamos, la metodología que proponemos en esta tesis, también incluye un proceso de extracción y uso de activos al cual llamamos *Recuperación*. Este proceso estará conformado por una serie de etapas que debe seguir el ingeniero para extraer los activos de requisitos del catálogo y utilizarlos en su nuevo proyecto.

El proceso de recuperación, según observamos en el panel derecho de la Figura 15, inicia con la *Consulta* del activo requerido para solucionar un problema específico. Luego se realiza el proceso de *Selección*, donde se analiza a los elementos obtenidos de la consulta para ver cuáles son los resultados relevantes respecto al objetivo de búsqueda, si no es satisfactoria la búsqueda se realiza nuevamente la consulta. Obtenido del paso de Selección el mejor activo para reusarlo, se realiza el proceso *Integración*, que consiste en adaptar la estructura de especificación del catálogo, a la estructura de especificación del nuevo proyecto. Finalmente, el activo pasa por el proceso de *Adecuación*, en el cual se adecua el contenido a la situación del nuevo proyecto y de esta manera terminar el proceso de *Recuperación*.

### 3.4.1 Indexación

En todo proceso de reúso, es necesario definir la forma de indexación de los activos reusables. En el caso de MORORE, definiremos la indexación como el proceso que va desde la identificación y preparación de los activos software reusables hasta su almacenamiento. El objetivo de este proceso de indexación es crear los índices necesarios que posibiliten la posterior búsqueda y recuperación de los activos software de manera eficiente.

Como ya lo mencionamos, para MORORE la *Indexación* empieza con la identificación del activo software (especificación del requisito potencialmente reusable según la plantilla de MORORE), el cual puede ser obtenido de un proyecto exitoso o

creado especialmente para ser reusado en múltiples proyectos. A continuación, el activo software identificado pasa por un proceso de control de calidad (la aplicación de los factores y métricas para medir la calidad del activo). Este control de calidad determina si el activo es descartado (no cumple con las condiciones mínimas para ser reusado), pasa a generalizarse (descripción clara, exacta y completa; parametrización, etc.) o si es óptimo para el reúso pasa a la etapa de *Indización* (obtener un índice del activo que facilite su búsqueda y recuperación posterior. Esta etapa puede ser automática o no, dependiendo de los recursos disponibles). Por último, es almacenado el activo y sus representaciones.

En esta sección, describimos detalladamente cada una de las etapas del flujo de indexación. Debemos señalar que el estilo de descripción usado fue inspirado en “El Cuerpo de Conocimientos de la Gestión de Proyectos (PMBok<sup>9</sup>: Project Management Body of Knowledge (PMBok, 2013)) del Instituto de Gestión de Proyectos (PMI<sup>10</sup>: Project Management Institute (PMI, 2017)). En cada una de las cinco etapas, proponemos productos de entrada, herramientas y técnicas, criterios de validación, y productos de salida que consideramos son básicos para el proceso. Se debe aclarar que, al momento de aplicar la metodología, el ingeniero debe evaluar qué elementos son relevantes, puesto que, son optativos y su utilidad está en función del objetivo del trabajo que se desea realizar.

### 3.4.1.1 Identificación

La identificación del activo de requisitos es el punto de partida en el proceso de indexación. Su objetivo es encontrar activos a partir de fuentes de conocimiento implícitas o explícitas.

Los activos de requisitos y en general del software no son descubiertos, sino creados. Cuando el ingeniero de requisitos revisa especificaciones de proyectos pasados, es capaz de reconocer activos que estuvieron muy bien elaborados. Es frecuente que estos activos especifiquen aspectos recurrentes en los proyectos de software y, por lo tanto, son potencialmente reusables. Después de documentar estos activos, es importante que el ingeniero complemente la información recopilada y describa a qué necesidades responden y en qué casos podrían ser utilizados.

Por otro lado, el activo también puede ser elaborado a partir de una necesidad, en este caso el punto de partida es el problema. El ingeniero identifica un problema recurrente en el proceso de especificación, lo define y busca entre las fuentes de información que tiene a su alcance las soluciones que otros han implementado de manera exitosa. Finalmente, llega a un consenso del análisis para obtener un activo específico que solucione de la mejor manera el problema.

En cualquiera de los dos casos mencionados el resultado debe ser el mismo, un activo de requisitos potencialmente reusable. Este activo debe estar conformado por una

---

<sup>9</sup> PMBoK: documento que contiene el conjunto de conocimientos y buenas prácticas aplicables a la gestión de proyectos bajo el enfoque del PMI.

<sup>10</sup> PMI: Asociación de profesionales de reconocimiento mundial dedicada a la investigación y promoción de las actividades de la gerencia de proyectos.

## SOLUCIÓN PROPUESTA

descripción suficientemente explícita para continuar el proceso de almacenamiento y posterior reúso. En la Figura 16 representamos el proceso de *Identificación* del activo, sus entradas, salidas, herramientas y técnicas, y criterios de validación.

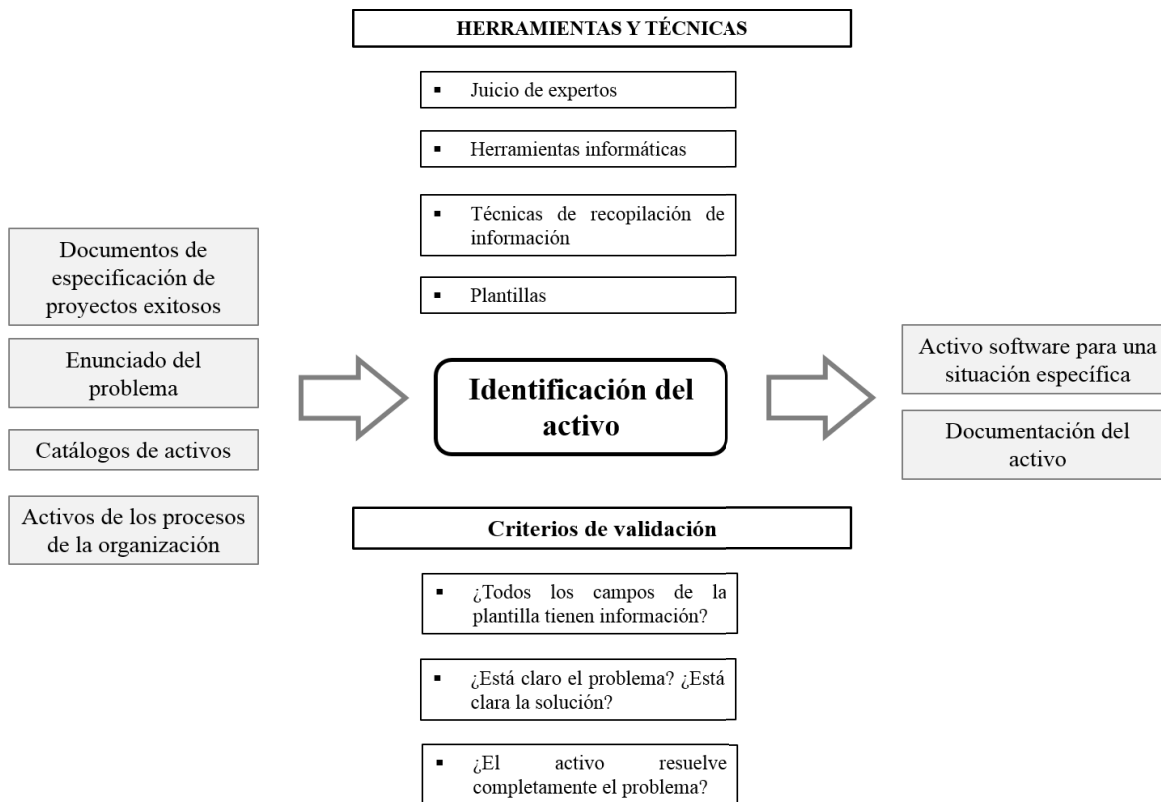


Figura 16. Etapa de Identificación del activo.

### 3.4.1.1.1 Entradas

#### a) Documentos de especificación de proyectos exitosos

Generalmente, los activos son identificados dentro de un proyecto de software, es decir, al momento de la elaboración de la especificación del sistema o cuando se revisa la especificación de un proyecto pasado. En estos documentos, los ingenieros, usuarios de la metodología, pueden identificar activos de requisitos que se repiten de un proyecto a otro con pequeñas variaciones. Éstos pueden ser seleccionados según diversos criterios (problema que resuelven, funcionalidad que especifican, etc.), escogiendo los más convenientes para reusarlos.

En este sentido, los documentos de especificación de proyectos exitosos son una buena fuente de activos de requisitos que debe ser revisada por el ingeniero para el proceso en mención.

#### b) Enunciado del problema

Ya mencionamos que también se puede partir del problema que los activos de requisitos van a solucionar. Una vez que conocemos el problema, buscamos la mejor solución en proyectos pasados o la preparamos basándonos en la experiencia.

## SOLUCIÓN PROPUESTA

---

El enunciado del problema es una descripción narrativa del problema o los problemas que se desean solucionar con el activo. Este enunciado debe ser elaborado a partir de situaciones repetidas que se han identificado en los proyectos de software anteriores. El enunciado del problema debe ser corto y preciso, debe definir con objetividad el problema que se desea solucionar y evitar las ambigüedades. De preferencia el enunciado debe describir solo un problema.

El enunciado del problema puede hacer referencia a funcionalidades específicas en los proyectos de software, una o varias restricciones del SDP, restricciones organizacionales comunes, etc.

### **c) Catálogos de activos**

Existen catálogos que han sido elaborados por organizaciones y grupos de investigación. Estos catálogos están alineados a los distintos estándares que existen y a las diversas metodologías de indexación. Usualmente, en estos catálogos los requisitos ya están generalizados y clasificados según diversos criterios.

### **d) Activos de los procesos de la organización**

Los activos de los procesos de la organización abarcan algunos o todos los activos de la misma que pueden utilizarse para mejorar el resultado del proceso de indexación. Entre estos activos tenemos:

- Lecciones aprendidas de proyectos pasados.
- Documentación e información histórica de la especificación de proyectos pasados.
- Conocimiento explícito de la organización.

#### **3.4.1.1.2 Herramientas y técnicas**

### **a) Juicio de expertos**

El juicio de expertos es una herramienta que consiste en recurrir a la experiencia de una o varias personas con conocimientos o capacitación especializada en un determinado tema. En esta etapa de identificación, a menudo se utiliza el juicio de expertos para evaluar las alternativas de solución disponibles y seleccionar la más conveniente para el problema recurrente a resolver. Por ejemplo, ingenieros experimentados, personal especializado de la organización, etc.

### **b) Herramientas informáticas**

Son herramientas automáticas que facilitan o simplifican el trabajo del ingeniero. Para esta primera etapa, se requiere aplicaciones que simplifiquen el proceso de llenado de la información de la plantilla del activo de requisitos de MORORE. Por ejemplo, las herramientas CASE.

### **c) Técnicas de recopilación de información**

Los ingenieros de software representan una fuente de información útil para complementar los esfuerzos realizados en la identificación y descripción de un activo.

## SOLUCIÓN PROPUESTA

---

Para acceder a su experiencia y sus conocimientos es necesario utilizar técnicas de recopilación de información que sean sistemáticas, objetivas y eficaces. Entre estas técnicas tenemos:

- Entrevistas: Una entrevista es una conversación dirigida con un propósito específico que utiliza un formato de preguntas y respuestas. Las entrevistas permiten al analista conocer las opiniones y experiencias de otros expertos.
- Cuestionarios y encuestas: Son conjuntos de preguntas escritas típicamente simples, diseñadas para acumular información repetitiva rápidamente, proveniente de un amplio número de encuestados.

### **d) Plantillas**

Las plantillas facilitan la presentación y síntesis de la información a través de una estructura predefinida. A pesar de que las plantillas se pueden adaptar, lo importante de esta herramienta es que ayuda al ingeniero a elaborar el activo de manera ordenada y estandarizada.

Para la presente metodología se ha propuesto una plantilla por cada activo que contiene cada uno de los elementos que describen al mismo. Hay que señalar que, para el activo del nivel de requisito individual, proponemos una plantilla configurable, pero típicamente hemos usado una plantilla estándar para los ejemplos creados.

#### **3.4.1.1.3 Criterios de validación**

Los criterios que den un mínimo de garantía para que la etapa de *Identificación* se considere completa, se especifican al responder las siguientes preguntas:

- ¿Todos los campos de la plantilla tienen información?
- ¿Está claro el problema? / ¿Está clara la solución?
- ¿El activo resuelve completamente el problema?

#### **3.4.1.1.4 Salidas**

### **a) Activo software para una situación específica**

Cuando la etapa de *Identificación* ha concluido, el resultado que se tiene es la plantilla del activo de requisitos con la información completa y correcta. En este punto, la plantilla todavía hace referencia a una situación específica, sin embargo, este patrón podría estar parcialmente generalizado. Lo importante en este punto es que todos los elementos de la plantilla contengan información correcta y que cumplan con el modelo de plantilla mostrado.

### **b) Documentación del activo**

La documentación del activo hace referencia a toda la información complementaria sobre el activo, que no se encuentra en la plantilla pero que puede ser útil para el desarrollador en las etapas posteriores. Esta información complementaria

## SOLUCIÓN PROPUESTA

pueden ser los documentos de donde se extrajo el activo, el contexto en el cual se está identificando el activo, los comentarios de otros ingenieros respecto al activo, etc.

### 3.4.1.2 Control de calidad

Después de haber identificado el activo de requisitos potencialmente reusable, es preciso evaluarlo. Este activo tiene que ser sometido a un proceso de control de calidad para garantizar que cumple con los requisitos mínimos para su almacenamiento y posterior reúso.

En esta fase, lo primero que debe hacer el ingeniero es analizar el modelo propuesto para medir la calidad. Adicionalmente, se debe aclarar los requisitos mínimos de calidad para los activos del catálogo, éste será el criterio de comparación para determinar si un activo está listo para almacenarse o si debe ser revisado y mejorado.

Lo que se debe hacer en esta etapa es inspeccionar el activo y evaluarlo de acuerdo al modelo de calidad establecido. Si el activo es aprobado (cumple las condiciones para el reúso), se puede continuar con la etapa de *Indización*; sin embargo, si el activo es desaprobado, se analiza si puede ser mejorado, en este caso, pasa la etapa de *Generalización*. En caso contrario, cuando el activo no cumple un mínimo de calidad para ser reusado, se descarta el activo se termina el proceso. En la Figura 17 representamos el paso de Control de Calidad del proceso de Indexación.

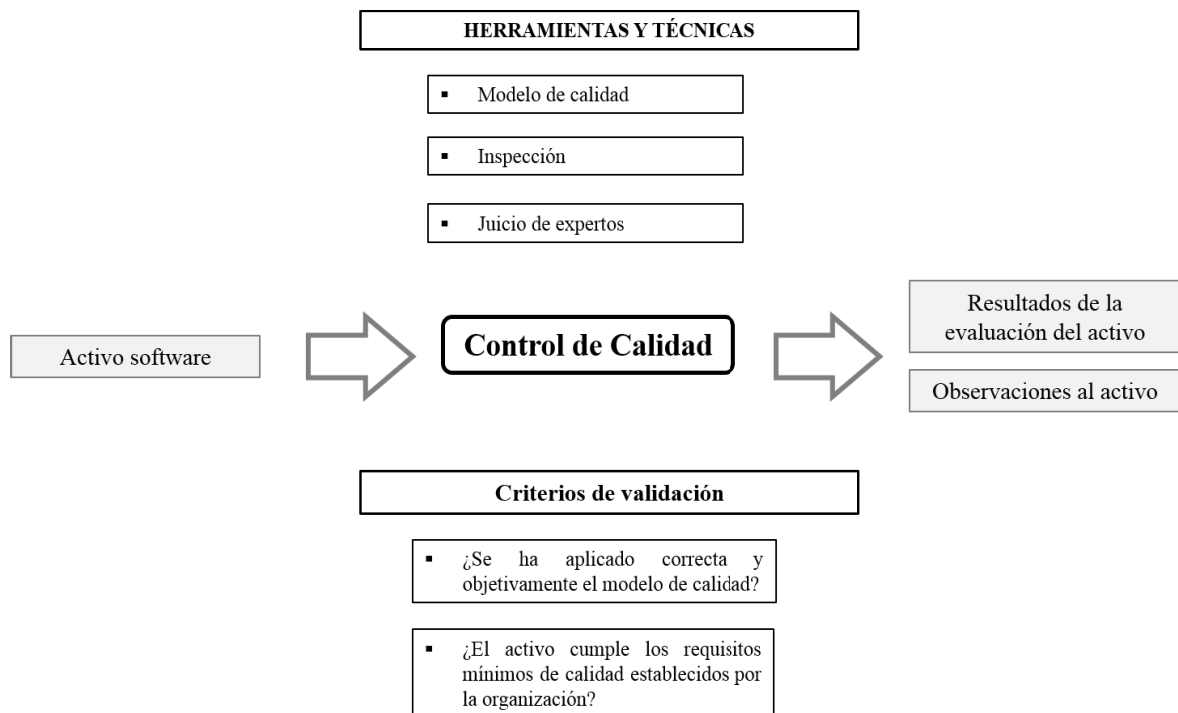


Figura 17. Etapa de Control de calidad.

### **3.4.1.2.1 Entradas**

#### **a) Activo Software**

En esta etapa, se le realiza un control de calidad al activo de requisitos que se obtuvo en una etapa previa. La evaluación puede realizarse también al diccionario del activo, si es el caso.

### **3.4.1.2.2 Herramientas y técnicas**

#### **a) Modelo de calidad**

Es una herramienta que define un método para evaluar la calidad de los activos. El modelo está conformado por: los factores de calidad (cualidades deseables), los indicadores de calidad (métricas) y las relaciones entre estos dos elementos. Todos estos componentes deben garantizar un control de calidad objetivo. Para esta metodología el modelo de calidad propuesto es el definido en el apartado 3.3 Modelo de calidad.

#### **b) Inspección**

La inspección es el examen del activo, incluye actividades como medir, examinar y verificar para determinar si el activo cumple con los requisitos de calidad definidos por el modelo. Esta inspección se realiza siguiendo los pasos establecidos por el modelo de calidad.

#### **c) Juicio de expertos**

El juicio de expertos se utiliza para evaluar el activo de requisitos y determinar su calidad. A pesar de que no es obligatorio que el activo sea revisado por un experto, su aporte sería muy enriquecedor si realiza el control de calidad. Otras de las ventajas son la objetividad y rigurosidad que una tercera persona aporta al proceso.

### **3.4.1.2.3 Criterios de validación**

- ¿Se ha aplicado correcta y objetivamente el modelo de calidad?
- ¿El activo cumple los requisitos mínimos de calidad establecidos por la organización?

### **3.4.1.2.4 Salidas**

#### **a) Resultados de la evaluación del activo**

La salida principal de esta etapa es la calificación del activo luego de ser sometido al control de calidad. Este resultado debe estar relacionado con una escala de medición que puede ser cuantitativa o cualitativa.

#### **b) Observaciones al activo**

Esta salida es importante en caso el activo no haya cumplido con los requisitos mínimos de calidad exigidos. En una lista se deben registrar las razones por las cuales el



## SOLUCIÓN PROPUESTA

activo ha sido descalificado. Esta lista será utilizada por el ingeniero para corregir los problemas de calidad presentes en el activo o para descartarlo.

### 3.4.1.3 Generalización

Este paso de la Indexación tiene como objetivo abstraer el activo para que pueda ser aplicado a cualquier situación. Se deben reemplazar los datos circunstanciales del activo, es decir, propios de una situación específica, por variables y generalizar la redacción del contenido. Además, se mejora la especificación según los resultados del control de calidad.

Un activo software es un conjunto de elementos relacionados entre sí. Estas relaciones son motivo para sugerir que la generalización tenga una visión global y no generar conflictos al tratar de forma independiente cada componente del activo. En la Figura 18 representamos el subproceso de Adaptación.

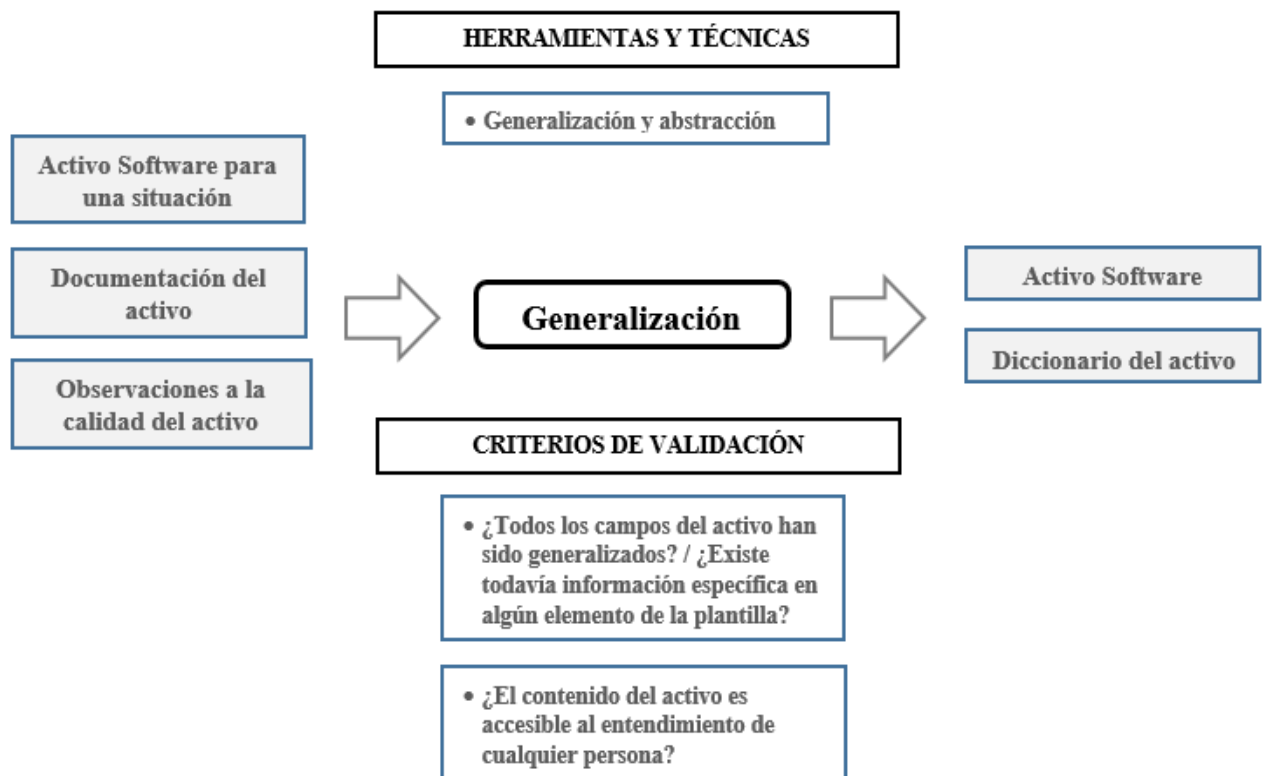


Figura 18. Etapa de Adaptación.

En esta etapa de *Generalización* proponemos cuatro sub pasos que debe seguir el ingeniero de requisitos para lograr una abstracción eficiente del activo.

- **Análisis:** lo primero que se debe hacer para adaptar el activo es analizar su contenido y depurar la información de “relleno”, es decir, aquella no relevante.
- **Clasificación:** en este paso el ingeniero debe separar la información general del activo de la información circunstancial. La información general del activo es constante en todas las realidades en las que el activo sea aplicado y constituye la esencia del mismo. Por otro lado, la información circunstancial es relativa y está relacionada a una situación en particular.

## SOLUCIÓN PROPUESTA

---

- **Parametrización:** consiste en el reemplazo de la información circunstancial por variables que puedan ser instanciadas al momento de reusar el activo. En este paso, el analista debe documentar todas estas variables en un diccionario, según se va a explicar posteriormente al tratar las salidas de este sub paso.

La *Generalización* tiene dos objetivos fundamentales:

- Editar el activo para mejorar la redacción y el contenido del mismo. En el paso anterior el activo ha quedado prácticamente generalizado y adaptado. Sin embargo, el analista debe revisar la ortografía y gramática; así como, mejorar la redacción siguiendo los lineamientos de calidad esperados para el activo.
- Verificar que el activo conserva su esencia. Una vez parametrizado y editado el activo, este debe ser revisado en su conjunto para verificar que en el proceso no se haya distorsionado su contenido. El activo debe seguir siendo una solución al problema planteado inicialmente.

### 3.4.1.3.1 Entradas

#### a) Activo software para una situación específica

Es la entrada principal para el proceso de *Generalización*. El ingeniero utilizará el activo de requisitos obtenido en la fase anterior como punto de partida y lo modificará sistemáticamente para obtener un activo generalizado y aplicable a cualquier situación.

#### b) Documentación del activo

Durante el proceso de generalización, el ingeniero debe considerar las condiciones y el contexto del activo original para no distorsionar la esencia del mismo. Esta información fue documentada en la etapa de identificación del activo.

#### c) Observaciones a la calidad del activo

Si el activo ya ha sido sometido a un proceso de control de calidad y su calificación final no ha sido la esperada, es necesario tener todas las observaciones en cuenta para poder mejorar el activo.

### 3.4.1.3.2 Herramientas y técnicas

#### a) Generalización y abstracción

La Adaptación del activo de requisitos puede realizarse por abstracción o por generalización. La abstracción prescinde de detalles accidentales y permite seleccionar solamente los aspectos fundamentales del activo modelado. Por otro lado, la generalización elige representar sólo los aspectos comunes a las diversas situaciones modeladas. Lo más conveniente para esta metodología es aplicar una combinación de ambos mecanismos.

### 3.4.1.3.3 Criterios de validación

- ¿Todos los campos del activo han sido generalizados? / ¿Existe todavía información específica en algún elemento de la plantilla?

## SOLUCIÓN PROPUESTA

- ¿El contenido del activo es accesible al entendimiento de cualquier persona?

### 3.4.1.3.4 Salidas

#### a) Activo software

El producto final de esta etapa es un activo de requisitos generalizado, es decir, que todos los elementos de información que componen el activo deben estar completos y parametrizados.

#### b) Diccionario del activo software.

Es un documento auxiliar generado por el proceso de parametrización. El diccionario del activo enumera todas las variables que han sido introducidas y proporciona una descripción más detallada de las mismas para facilitar su entendimiento al momento de ser reutilizado.

Proponemos el esquema de la Tabla 132 para representar el diccionario de un activo de requisitos.

Nº	Variable	Nombre	Descripción	Ejemplo
1	<#Us>	Número de usuarios	Cantidad de tipos de usuario que tendrá el sistema.	3
2	<NomUs>	Nombre de usuario	Formato/estructura de los nombres de usuario.	nombre.apellido

Tabla 132. Diccionario del activo software

### 3.4.1.4 Indización

La *Indización* es la etapa en la cual del activo software se obtiene un modelo de representación de la información, que permitirá su almacenamiento y posteriormente una óptima recuperación del repositorio. Este modelo de representación de la información estará conformado por dos elementos fundamentales: el fondo documental (los activos propiamente dichos) y el índice (modelos de representación).

El primer elemento, son todos los documentos que están registrados en el repositorio. En este caso específico, estos documentos son los activos de requisitos obtenidos en la etapa de *Generalización* y/o estandarizados según la plantilla.

El segundo elemento es el índice que ordenará el fondo documental y facilitará la búsqueda de sus elementos. Este índice debe estar conformado por un conjunto de elementos que describan el contenido de cada documento.

En esta etapa de *Indización*, el desarrollador se encargará de relacionar el activo que está trabajando con el índice a partir de un conjunto de términos de indización (Nombre, Descripción y Palabras clave del activo). El proceso puede ser realizado a través de métodos algorítmicos automatizados en un software o de modo intelectual. Esta elección dependerá directamente de los recursos que se tengan disponibles. La Figura 19 representa el sub paso de Indización del proceso de *Indexación*.

## SOLUCIÓN PROPUESTA

---

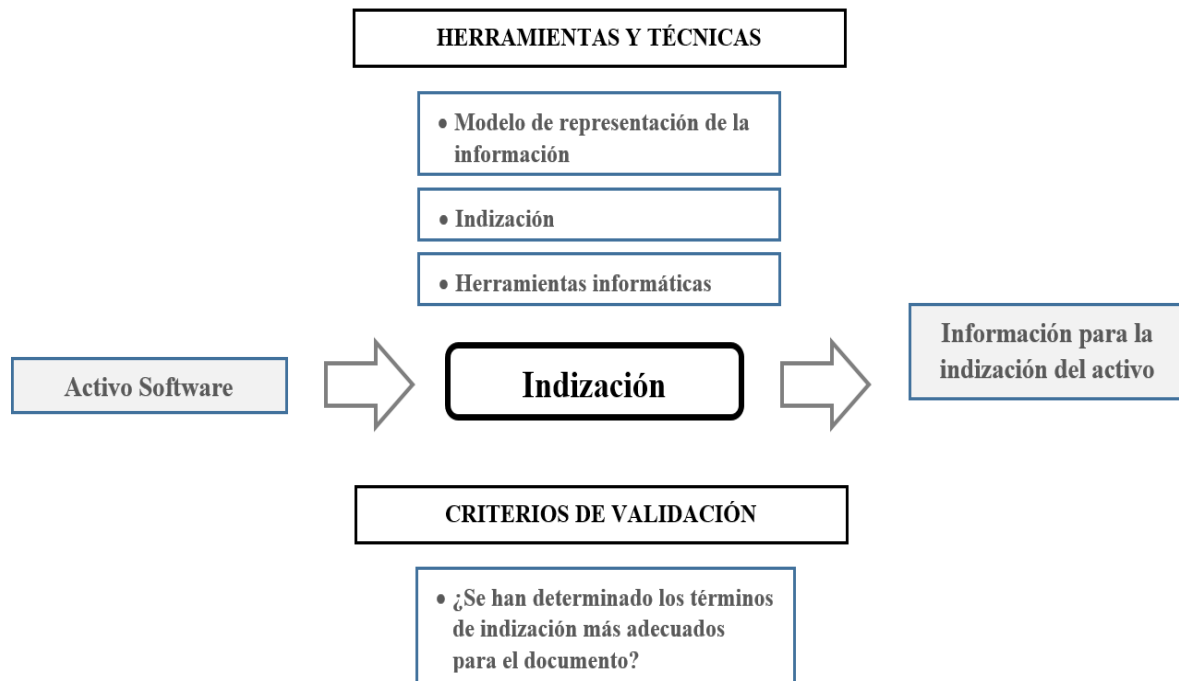


Figura 19. Etapa de Transformación.

### 3.4.1.4.1 Entradas

#### a) Activo Software

Es la información que se va a indizar. Se obtiene en la etapa de Generalización y está estructurada en una plantilla según como se va a almacenar en el repositorio.

### 3.4.1.4.2 Herramientas y técnicas

#### a) Modelo de representación de la información

El modelo de representación de la información es un modelo conceptual que describe la estructura general, las funciones básicas y los aspectos lógicos de una determinada forma de representar la información. Básicamente los modelos están orientados a la identificación de los conceptos más relevantes del activo para determinar la lista de términos de indización.

Existen diversos modelos y algoritmos de indización. En el caso de que este proceso se realice de manera automática con la ayuda de un software especializado, el modelo será transparente al ingeniero y estará embebido en la herramienta.

#### b) Indización

La indización se divide en dos pasos: análisis y normalización. El análisis consiste en la identificación de los temas o características semánticas más relevantes del activo. La normalización, transforma los conceptos obtenidos en un conjunto de términos descriptores más adecuados. Cada uno de estos términos normalizados sobre qué trata el activo será una entrada al índice que facilite la consulta y recuperación.

### c) Herramientas informáticas

A pesar de que un proceso de indización inteligente es más eficiente que uno automatizado, cada día los resultados obtenidos con herramientas de software son más acertados. En función de la complejidad del software, puede automatizarse completamente el proceso o complementarse con acciones externas realizadas por el analista. En cualquiera de los casos, el soporte informático simplificará el trabajo del ingeniero.

#### 3.4.1.4.3 Criterios de validación

- ¿Se han determinado los términos de indización más adecuados para el documento?

#### 3.4.1.4.4 Salidas

### a) Información para la indización del activo

La información para la indización del patrón son un conjunto de palabras que servirá para relacionar el activo con las consultas de los usuarios y determinar si debe ser un resultado relevante de una búsqueda específica.

La información se puede representar como un vector de “n” posiciones, donde cada posición contiene un término que puede ser o bien palabras simples (“ingeniería”) o bien compuestas (“ingeniería industrial”) que expresan las propiedades semánticas del documento.

#### 3.4.1.5 Almacenamiento

La última etapa del proceso de *Indexación* es el almacenamiento. Esta etapa tiene como objetivo guardar la información para que esté disponible en el futuro. Dependiendo de los procesos, el almacenamiento puede ser en un archivo físico o en una base de datos.

Un archivo físico es una alternativa para una organización pequeña que almacenará una cantidad de activos reducida. En este caso, la persona encargada debe tener conocimientos de archivología que garantice consultas rápidas al archivo y la conservación de los documentos.

Definitivamente, el almacenamiento digital es el más conveniente, porque a través de los sistemas informáticos automatizados es posible simplificar los procesos de búsqueda, recuperación y actualización de los archivos. Adicionalmente, los repositorios digitales facilitan la distribución y difusión de la información cuando se requiera. La Figura 20 representa el sub paso de *Almacenamiento* del proceso de *Indexación*.

## SOLUCIÓN PROPUESTA

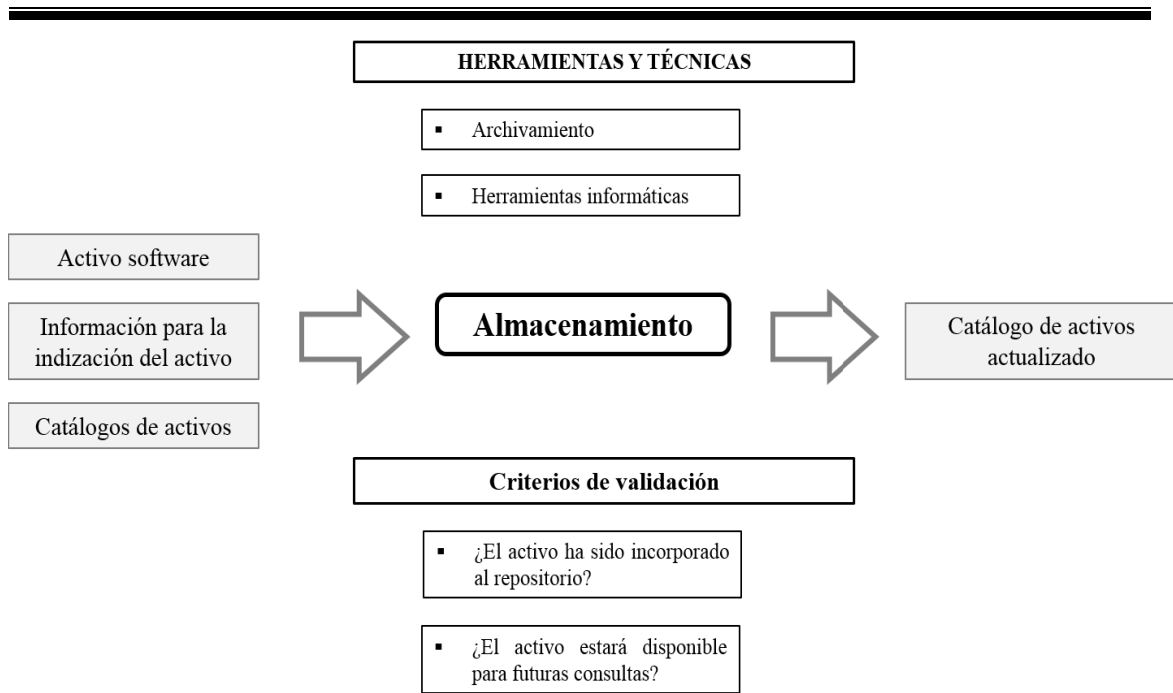


Figura 20. Etapa de Almacenamiento.

### 3.4.1.5.1 Entradas

#### a) Activo software

Es el activo almacenado bajo una estructura estandarizada (plantilla) que contiene la información y experiencia del ingeniero senior, para que pueda ser reusada por otros ingenieros nóveles. Este activo fue obtenido en la etapa de Generalización y debe ser almacenado junto con su diccionario.

#### b) Información para la indización del activo

Son el conjunto de elementos que describen el activo. Estos elementos son almacenados en relación al activo para facilitar la selección del mismo al momento de una consulta. Cada uno de los elementos será una entrada al índice del catálogo.

#### c) Catálogo de activos

Cada nuevo activo debe ser indexado en un repositorio. Este repositorio es un catálogo de activos de requisitos que puede ser público o privado; físico o digital.

### 3.4.1.5.2 Herramientas y técnicas

#### a) Archivamiento

Las técnicas de archivo son los métodos teóricos y prácticos para el almacenamiento seguro del catálogo y la recuperación de los documentos físicos. En el caso de esta metodología, los archivos que se van a almacenar son los activos de requisitos y pueden ser en formato físico o digital. Para almacenar los documentos, se deben realizar tres operaciones: clasificación, ordenación e instalación.

### b) Herramientas informáticas

La automatización de este proceso a través de un software implica el almacenamiento de los activos en una base de datos documental. El almacenamiento electrónico ofrece muchas ventajas respecto al almacenamiento físico en términos de practicidad, conservación de la información y eficiencia al momento de las consultas.

#### 3.4.1.5.3 Criterios de validación

- ¿El activo ha sido incorporado al repositorio?
- ¿El activo estará disponible para futuras consultas?

#### 3.4.1.5.4 Salidas

### a) Catálogo de activos actualizado

Al final del proceso de *Indexación* de un activo software, el resultado es el catálogo con nuevos activos incorporados.

### 3.4.2 Recuperación

Para MORORE, el proceso de recuperación consiste en la extracción efectiva de los activos de requisitos almacenados, para su reuso efectivo en una situación específica.

Este proceso comienza con la consulta de activos relacionados a la situación o problema específico del ámbito de los requisitos. Esta consulta permite la extracción de los activos mediante búsquedas semánticas, posibles, gracias a la indexación descrita anteriormente. Esta indexación conjuntamente con la correcta especificación del problema y criterios de búsqueda, permitirá una efectiva recuperación, mejorando los índices de medida de la efectividad de la recuperación (Precisión y Exhaustividad<sup>11</sup>). Luego se pasa al proceso de *Selección* que consiste en estudiar y escoger el o los activos de requisitos que más se aproximen a los requerimientos de una situación específica. Posteriormente, estos activos de requisitos seleccionados se extraen e integran en el proyecto específico donde serán reusados. Esta integración consiste en adecuar la estructura de especificación del activo a la estructura de especificación de requisitos del nuevo proyecto. Por último, se adecua (amoldar el contenido textual del activo al nuevo proyecto) el activo al proyecto específico y termina el proceso de reuso.

En esta sección, se describirá el proceso de *Recuperación* de un activo de requisitos a partir de una necesidad. Para la descripción de cada etapa utilizaremos la misma estructura que usamos en la sección anterior.

#### 3.4.2.1 Consulta

El objetivo de esta primera etapa es encontrar, a partir de la consulta al catálogo, todos los activos almacenados que podrían ser de utilidad en el nuevo proyecto

---

<sup>11</sup> La *Precisión*, se obtiene de la división de la cantidad de elementos relevantes recuperados, entre el total de elementos recuperados. Y la *Exhaustividad*, se obtiene de la división de elementos relevantes recuperados, entre el total de elementos relevantes existentes en el repositorio.

## SOLUCIÓN PROPUESTA

(“Exhaustividad máxima”). Y si ante la consulta, no se obtienen activos inadecuados al nuevo proyecto se logra una “Precisión máxima”.

Es importante recalcar que, en esta etapa, el ingeniero no debe analizar, comparar ni calificar los resultados obtenidos. Lo único que debe hacer el ingeniero es una revisión superficial para descartar los resultados que no sean relevantes. La Figura 21 representa el paso de *Consulta* del proceso de *Recuperación*.

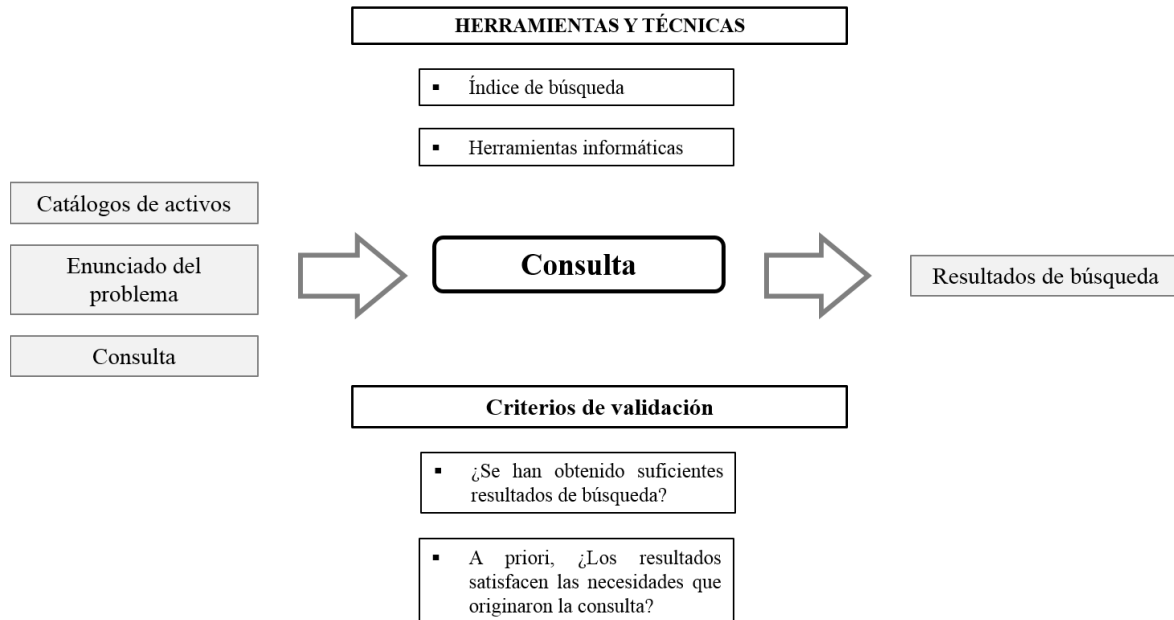


Figura 21. Etapa de Consulta.

Para empezar el proceso, el ingeniero debe identificar, en el proyecto que está trabajando, las oportunidades de reúso que se presentan. Una vez que tiene claro su objetivo de búsqueda, debe proceder con la consulta al catálogo.

Durante la búsqueda en el catálogo se va a extraer el conjunto de activos más relevante para la necesidad del usuario. Esta se puede realizar de manera manual o con la ayuda de un software.

### 3.4.2.1.1 Entradas

#### a) Catálogo de activos

El catálogo de activos es el repositorio que almacena un conjunto de activos de requisitos para que puedan ser reusados. Este catálogo puede ser físico o digital; privado o público.

#### b) Enunciado del problema

Al igual que en la *Indexación*, el enunciado del problema es una descripción narrativa de la necesidad. Para elaborar el enunciado del problema, el analista debe primero bosquejar los requisitos que tendrá el nuevo sistema y ordenarlos según sus características y relaciones. Luego, se recomienda identificar aquellos elementos que



## SOLUCIÓN PROPUESTA

---

podrían ser obtenidos a partir del repositorio. Finalmente, se debe redactar en palabras sencillas el objetivo de la búsqueda.

Si durante el proceso se identifican varias oportunidades de reúso, es conveniente que se separen en grupos y se prepare un enunciado para cada uno.

### c) **Consulta**

La Consulta es una frase corta que sintetiza el enunciado del problema y que constituirá el criterio de búsqueda del activo. Las palabras más relevantes de la consulta son comparadas con la información indizada del activo, obtenida en la etapa de *Indización* del proceso de *Indexación*.

#### 3.4.2.1.2 *Herramientas y técnicas*

### a) **Índice de búsqueda**

El índice de búsqueda es un documento análogo a un diccionario y permite al analista buscar los activos que más se ajustan a sus necesidades. Este índice debe estar ordenado de tal manera que permita consultas rápidas y eficientes.

El ingeniero debe utilizar el índice para ubicar y luego extraer los activos que podrían ser útiles en su proyecto actual. Este índice es muy importante para los repositorios físicos o digitales que no tienen un sistema de búsqueda automatizado.

### b) **Herramientas informáticas**

Los repositorios digitales que son gestionados mediante una aplicación web o de escritorio. Permiten realizar consultas automatizadas que facilitan el trabajo del ingeniero.

Dependiendo de la complejidad de la aplicación, este tipo de software puede estar basado en distintos métodos de búsqueda, siendo lo más recomendable la búsqueda semántica basada en un tesoro. Un tesoro es una herramienta para el control del vocabulario semántico que orienta a los indizadores y a los usuarios sobre los términos que pueden utilizar y, así ayuda a mejorar la calidad de la recuperación.

#### 3.4.2.1.3 *Criterios de validación*

- ¿Se han obtenido suficientes resultados de búsqueda?
- A priori, ¿Los resultados satisfacen las necesidades que originaron la consulta?

#### 3.4.2.1.4 *Salidas*

### a) **Resultados de búsqueda**

En esta etapa el resultado aún no son los activos en sí, sino una lista de activos que podrían ser útiles al ingeniero. La lista podría contener simplemente el nombre, el identificador del activo y los términos de indización; o ser un poco más completa y contener un pequeño resumen de cada uno.

# SOLUCIÓN PROPUESTA

## 3.4.2.2 Selección

En el paso de *Selección*, se debe evaluar cada uno de los resultados obtenidos en la consulta, para escoger los que se van a utilizar en el proyecto. Esta es la etapa de análisis propiamente dicha y debe considerar principalmente 2 criterios:

- Eficacia. El activo debe ser una solución al problema que ha identificado el ingeniero.
- Compatibilidad. Cada activo demanda requisitos previos y restricciones que deben ser contrastadas con el proyecto para comprobar que es factible su incorporación.

Durante la selección, el ingeniero puede percatarse de que los resultados obtenidos no son los adecuados o no son suficientes, en ese caso, debe realizar nuevamente una consulta para obtener nuevos resultados que complementen los que ya tiene. La Figura 22 representa el paso de *Selección* del proceso de *Recuperación*.

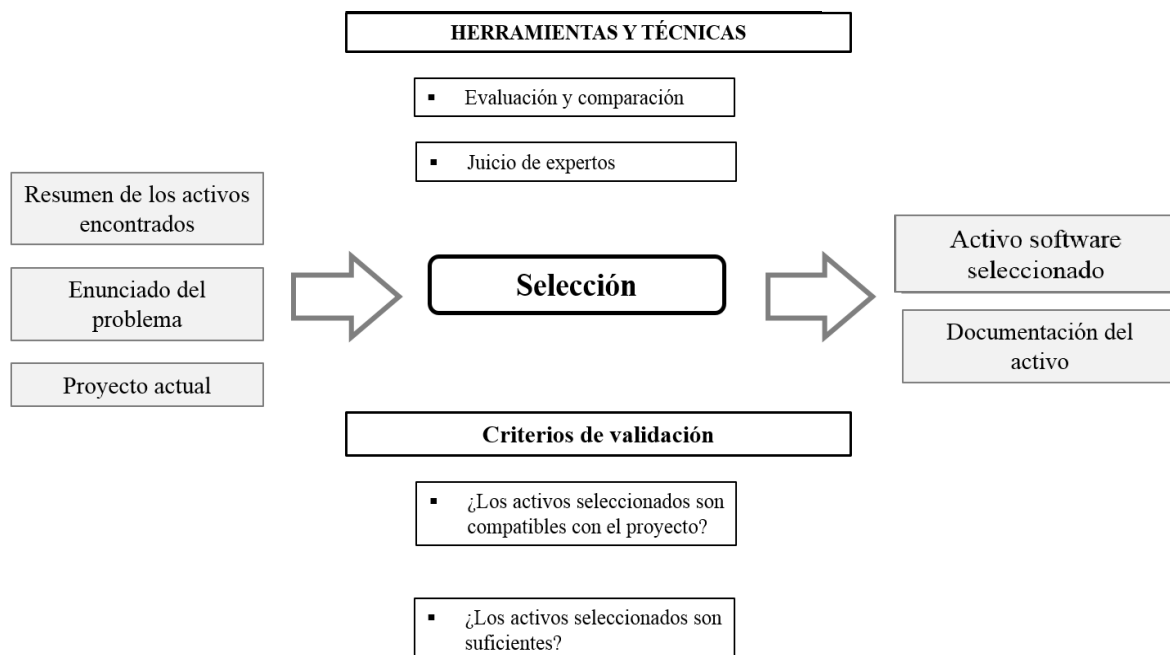


Figura 22. Etapa de Selección.

### 3.4.2.2.1 Entradas

#### a) Resumen de los activos encontrados

No es conveniente que el ingeniero revise cada uno de los activos por completo porque sería un trabajo muy pesado y demandaría demasiado tiempo. Lo más oportuno es que el análisis se realice sobre un resumen del activo en el cual se sinteticen los aspectos más relevantes del mismo.

La información del resumen puede ser obtenida de la plantilla o de información complementaria almacenada en el catálogo. En esta fase, los puntos más relevantes de la plantilla son:

- Nombre
- Descripción

### **b) Enunciado del problema**

El enunciado fue elaborado antes de hacer la consulta y especifica objetivamente las necesidades del ingeniero. En esta actividad, el enunciado será una referencia de las funcionalidades y aspectos del software que se esperan cubrir con los activos.

### **c) Proyecto actual**

Los activos que se seleccionen van a formar parte del proyecto, por lo tanto, es importante que el ingeniero en todo momento tenga presente el proyecto y en especial sus características más relevantes. La evaluación de cada uno de los activos debe estar alineada con el proyecto en el cual se van a acoplar.

#### **3.4.2.2.2 Herramientas y técnicas**

### **a) Evaluación y comparación**

Para seleccionar el activo que más conviene al proyecto es necesario, en primer lugar, evaluar las condiciones y restricciones de los activos obtenidos (*Descripción*). Esta evaluación permitirá al ingeniero descartar aquellos activos que no son compatibles con el proyecto.

Cuando se tengan los activos compatibles con el proyecto, se debe escoger el que mejor se adapte a las necesidades del mismo. En ese sentido el ingeniero debe comparar los activos para determinar el que más le conviene.

### **b) Juicio de expertos**

En la ingeniería de requisitos la experiencia del ingeniero es muy importante para aumentar las probabilidades de éxito del proyecto. Ante alguna disyuntiva entre los activos obtenidos, un ingeniero puede solicitar el apoyo de otro con mayor experiencia para seleccionar el activo que más le conviene.

#### **3.4.2.2.3 Criterios de validación**

- ¿Los activos seleccionados son compatibles con el proyecto?
- ¿Los activos seleccionados son suficientes?

#### **3.4.2.2.4 Salidas**

### **a) Activo software seleccionado**

Cuando el proceso de Selección está listo, el resultado es un activo que satisface las necesidades del proyecto y ha sido escogido por el ingeniero para incorporarlo al proyecto.

### **b) Documentación del activo**

La documentación del activo hace referencia a toda la información complementaria sobre el activo (que puede ser extraída con el activo), que no se encuentra

## SOLUCIÓN PROPUESTA

en la plantilla pero que puede ser útil para el ingeniero en las etapas siguientes de *Integración y Adecuación*.

### 3.4.2.3 Integración

Esta tercera etapa del proceso de recuperación consta de dos actividades consecutivas. La primera es la recuperación del activo propiamente dicha y la segunda es la integración del activo en el proyecto.

Cuando un activo es indexado, se adapta a un formato o esquema (plantilla) que es estándar para el repositorio de destino. La recuperación consiste en extraer del catálogo el activo con todos los campos de información completos, manteniendo la estructura de especificación con la que fue almacenado (esquema o plantilla estándar del repositorio).

Como cada proyecto es único y las organizaciones tienen su propia metodología para documentarlo, es posible que la estructura de especificación del activo no sea la misma que la del proyecto. A pesar de que se tendrían dos estructuras distintas, se espera que exista cierta correspondencia entre ambas. La integración consiste precisamente en trasladar el contenido del activo del formato con el que fue almacenado, al formato del proyecto, relacionando sus elementos análogos. Con una integración adecuada, se espera que el activo sea congruente con los demás elementos del proyecto. La Figura 23 representa el sub paso de Integración del proceso de Recuperación.

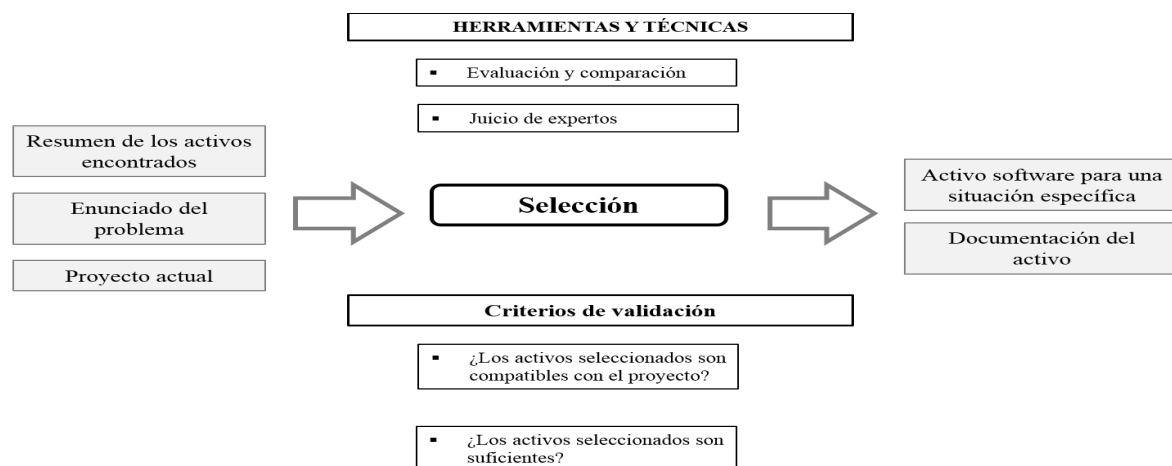


Figura 23. Etapa de Integración

#### 3.4.2.3.1 Entradas

##### a) Activo software seleccionado

Es el activo que ha sido escogido por el ingeniero para incorporarlo al proyecto. Este activo conserva la estructura de almacenamiento propia del catálogo del cual fue obtenido.

##### b) Proyecto actual

El activo seleccionado va a ser introducido en el documento de especificación del proyecto. El ingeniero no solo debe disponer del documento de especificación, sino

## SOLUCIÓN PROPUESTA

también, de los lineamientos establecidos por la organización para la elaboración del mismo.

### 3.4.2.3.2 Herramientas y técnicas

#### a) Plantillas y formatos

En la fase de recuperación e integración, las plantillas que se deben utilizar son las del proyecto nuevo. Cada proyecto es único y la estructura de su documentación varía de acuerdo a los estilos de la organización y del equipo de trabajo.

### 3.4.2.3.3 Criterios de validación

- ¿El activo se ha integrado al proyecto?

### 3.4.2.3.4 Salidas

#### a) Activo software integrado al proyecto

Es la sección del documento de especificación del sistema en el cual se encuentra la información del activo que se está reusando.

### 3.4.2.4 Adecuación

La *Adecuación* es la etapa en la cual el activo genérico del catálogo es ajustado a las características específicas del proyecto de software que se está desarrollando. El objetivo de esta fase es obtener la especificación que se había planteado en el enunciado del problema y que motivó la aplicación del reúso en el proyecto. La Figura 24 representa el paso de *Adecuación* del proceso de *Recuperación*.

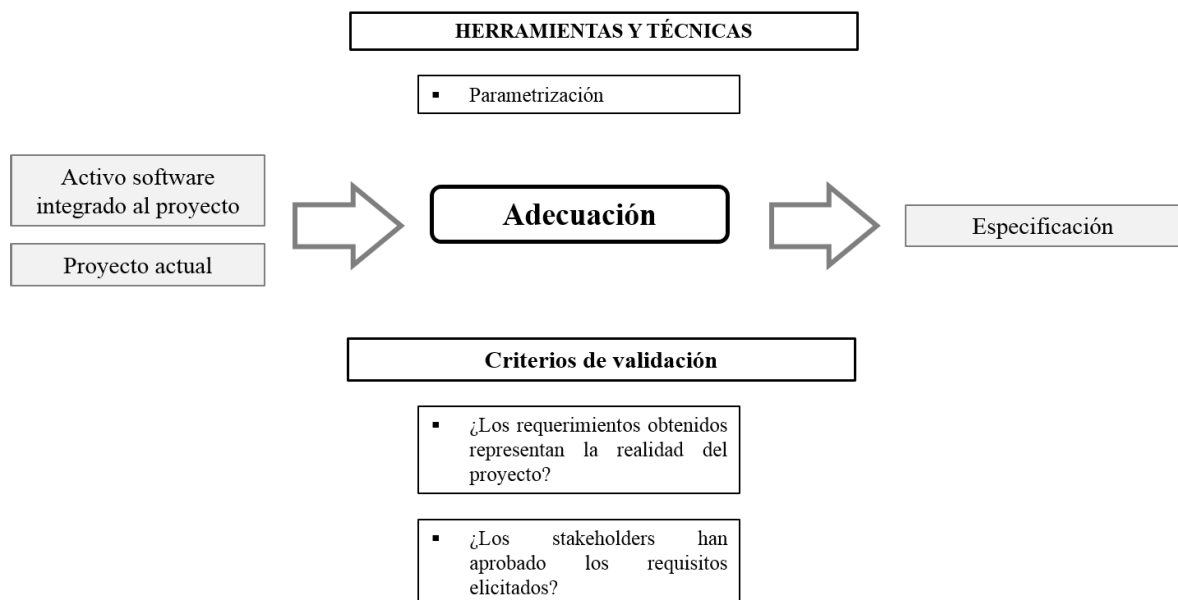


Figura 24. Etapa de Adecuación.

En la etapa anterior, el activo se incorporó al proyecto nuevo y se modificó su estructura de especificación para que esté en armonía con todo el documento de

## SOLUCIÓN PROPUESTA

---

especificación de requisitos, pero aún el contenido es genérico. En esta etapa, el ingeniero debe ajustar el activo para que represente la realidad del proyecto en el cual está trabajando. Cada una de las variables del activo debe asumir el valor del atributo que representa en el proyecto. Cuando la adecuación esté completa, el activo solo hará referencia al proyecto específico.

### **3.4.2.4.1 Entradas**

#### **a) Activo software integrado al proyecto**

El activo de requisitos integrado al proyecto es el resultado de la etapa anterior. Consiste en una sección del documento de especificación del proyecto en la cual se ha introducido el activo de requisitos que aún es general.

#### **b) Proyecto actual**

En ningún momento el ingeniero debe desligar el proyecto en el que está trabajando y el proceso de reúso, porque estos dos están estrechamente relacionados.

El proyecto involucra la realidad que se está modelando, la organización donde se está desarrollando, las personas involucradas, las metodologías aplicadas y la documentación disponible.

### **3.4.2.4.2 Herramientas y técnicas**

#### **a) Parametrización**

Esta herramienta consiste en instanciar cada una de las variables del activo. Debemos asignar un valor concreto a cada variable en función de los parámetros y las características del software que se está desarrollando en el proyecto nuevo.

Adicionalmente, debemos revisar la redacción total del activo instanciado, debido que durante la parametrización algunas frases se pueden ver afectadas y tengan la necesidad de un ajuste.

### **3.4.2.4.3 Criterios de validación**

- ¿Los requisitos obtenidos representan la realidad del proyecto?
- ¿Los stakeholders han aprobado los requisitos elicitados?

### **3.4.2.4.4 Salidas**

#### **a) Especificación**

La salida de la etapa de *Adecuación* es un conjunto de activos de requisitos para el proyecto que está desarrollando el ingeniero. Al final de este proceso, el resultado es la especificación de una funcionalidad o un aspecto del sistema en el que se ha aplicado el desarrollo con reúso.

### 3.4.3 Aplicación de la metodología

En las secciones anteriores explicamos cada uno de los procesos de esta metodología. En cada sub proceso de indexación y recuperación, enumeramos una serie de entradas, herramientas y técnicas, criterios de validación y salidas de subproductos. Sin embargo, reiteramos que los elementos son optativos y la metodología no exige que todos sean utilizados.

En esta sección vamos a indexar un activo de requisitos y paso a paso comentaremos las acciones realizadas para que quede más claro el trabajo que debe realizar el ingeniero usuario de esta metodología. Posteriormente, describiremos a través de otro ejemplo el proceso que debemos seguir para recuperar un activo almacenado en un catálogo.

Debido a las limitaciones de recursos al momento de realizar el presente trabajo de trabajo de investigación, estos ejemplos los desarrollaremos en su totalidad de manera manual. Sin embargo, esto no descarta la posibilidad que tiene el usuario de aprovechar herramientas informáticas automatizadas para agilizar el proceso descrito. Asimismo, el ejemplo está desarrollado para uno de los niveles de activos de requisitos de MORORE: los Patrones de Requisitos.

#### 3.4.3.1 Indexación

Para empezar con el ejemplo, necesitábamos un patrón de requisitos con el cual trabajar. Por este motivo, decidimos revisar documentos de especificación de software, para identificar un patrón que nos sirva para ilustrar la metodología.

##### 3.4.3.1.1 Identificación del activo

En este proceso inicial, nuestras entradas fueron los *Documentos de especificación de proyectos exitosos* que encontramos en nuestra revisión. Después de explorar varios proyectos de software, seleccionamos el documento técnico de Especificación de Requisitos para el desarrollo del “Sistema Integrado de Información Universitaria (SIIU)” (SIIU, 2010).

Este proyecto de la Secretaría General de Universidades del Ministerio de Educación, Cultura y Deporte de España, tuvo como objetivo disponer de indicadores del Sistema Universitario Español que sean de calidad, fiables y que permitan la comparabilidad. Además, el sistema informático previsto debía aportar la información necesaria a todos los agentes del sistema: Ministerio, Comunidades Autónomas (CCAA), Universidades, Estudiantes, Profesores e Investigadores, etc.

Elegimos este documento porque en el punto 3.1. *Requisitos Funcionales*, encontramos un potencial patrón de requisitos que se ajustaba a lo requeríamos. Este conjunto de requisitos identificado especificaba los perfiles y privilegios para los usuarios del software. Dicho patrón fue el activo de requisitos que usamos para el presente ejemplo. Los requisitos que escogimos en el documento para que formen parte de nuestro patrón los extrajimos y ordenamos en la Tabla 133.

## SOLUCIÓN PROPUESTA

Requisitos															
R01	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará el organismo al que pertenece y el nivel de actuación de su organización, ya sea Universidad, Comunidad Autónoma, Ministerio de Educación y otras Instituciones.														
R02	Los privilegios dados a un usuario, vendrán informados en el perfil que se asigne al usuario.														
R03	Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios: <ul style="list-style-type: none"><li>Tipos de Usuario (R04)</li><li>Niveles de Usuario (R05)</li></ul>														
R04	<p>Para limitar el acceso a los apartados que se definan en la aplicación existirán cuatro tipos de usuario:</p> <ul style="list-style-type: none"><li>Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general.</li><li>Ejecución. Además de poseer los permisos del usuario de Lectura, podrá acceder a la aplicación para ejecutar y visualizar informes más especializados, previamente desarrollados por otro perfil de usuario. En este perfil estarán, por ejemplo, el Observatorio Universitario o la Fundación Universidad.es.</li><li>Desarrollo. Además de poseer los permisos del usuario de Ejecución, podrá acceder a la aplicación para generar, ejecutar y visualizar informes. En este perfil estarán, por ejemplo, las Universidades, la ANECA, las CCAA.</li><li>Administración. Además de poseer los permisos del usuario de Desarrollo, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado del Ministerio de Educación, de las CCAA y de las Universidades.</li></ul>														
R05	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz.</p> <table><tr><td></td><td>Detalle del dato (microdato)</td><td>Datos agregados</td><td>Indicadores</td></tr><tr><td>Área Académica</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área Económica</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr></table>				Detalle del dato (microdato)	Datos agregados	Indicadores	Área Académica	Nivel	Nivel	Nivel	Área Económica	Nivel	Nivel	Nivel
	Detalle del dato (microdato)	Datos agregados	Indicadores												
Área Académica	Nivel	Nivel	Nivel												
Área Económica	Nivel	Nivel	Nivel												



## SOLUCIÓN PROPUESTA

	Área de Recursos Humanos	Nivel	Nivel	Nivel
	Área de Inserción Laboral	Nivel	Nivel	Nivel
	Área I+D	Nivel	Nivel	Nivel
<p>Dicha matriz deberá contemplar las distintas áreas (Académica, Económica, de Recursos Humanos, de Inserción Laboral e I+D) que existan en el sistema.</p> <p>El nivel de usuario indicado en la matriz podrá tomar los siguientes valores:</p> <ul style="list-style-type: none"> <li>• Nivel 4. No se tendrá acceso a este tipo de información.</li> <li>• Nivel 3. Indica que el usuario sólo podrá acceder a los datos de la Universidad a la que esté asociado.</li> <li>• Nivel 2. Indica que el usuario sólo podrá acceder a los datos de la Comunidad Autónoma a la que esté asociado, lo que implica que tendrá acceso a los datos de las Universidades que se localizan en dicha Comunidad Autónoma.</li> </ul> <p>Nivel 1. Corresponde a los usuarios que tienen el privilegio de acceder a todos los niveles de información.</p>				
R06	Se asignarán tipo de usuario con sus respectivos niveles de seguridad a todos los organismos implicados: Ministerio de Educación, CCAA, Universidad, ANECA, CRUE, Observatorio Universitario de Becas y Ayudas al Estudio y Rendimiento Académico, Fundación Universidad.es, etc.			

*Tabla 133. Requisitos del SIIU*

Con el patrón identificado, utilizamos dos de las *herramientas y técnicas* que propone la metodología: *Juicio de expertos* y *Plantillas*.

En primer lugar, analizamos el documento técnico de Especificación de Requisitos del SIIU y los requisitos que habíamos identificado en el proyecto para evaluar si formaban un buen patrón. Efectivamente, concluimos que esos requisitos constituían una propuesta muy interesante para la especificación de perfiles de usuario y su reuso como patrón sería muy útil para muchos proyectos.

Adicionalmente decidimos, para facilitar el entendimiento del patrón, desarrollar un Diagrama Conceptual partiendo de los requisitos y lo adjuntamos al patrón para que los usuarios puedan consultarlo.

## SOLUCIÓN PROPUESTA

Para el diagrama consideramos que era suficiente trabajar únicamente con las clases y omitir los atributos y métodos. El resultado podemos observarlo en la Figura 25.

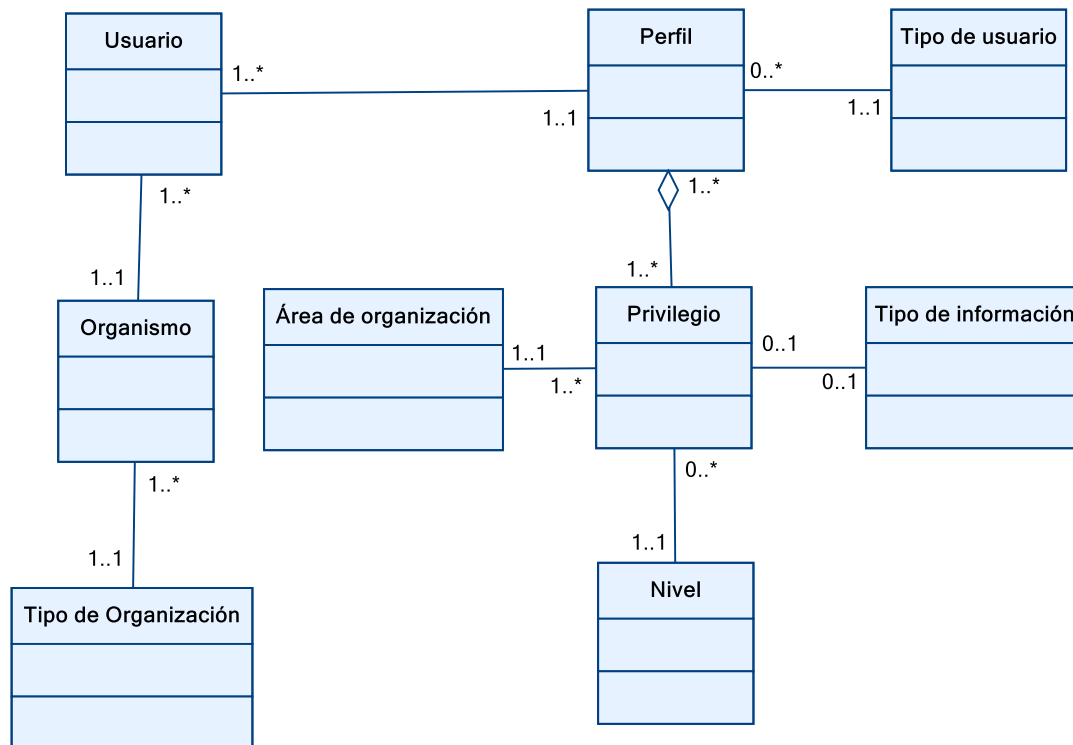


Figura 25. Diagrama conceptual del patrón.

Luego de analizar y entender los requisitos del patrón, procedimos a completar la plantilla que propusimos en esta metodología. En nuestro patrón ejemplo hemos definido los atributos que faltan de la siguiente manera:

- Identificador: PR0002
- Clasificación: Acceso al sistema
- Nombre: Perfil de usuario del sistema
- Descripción: Especifica los requisitos para definir los perfiles de acceso de los usuarios del sistema. Los perfiles limitan las acciones que podrán realizar los usuarios sobre el sistema y el grado de detalle de la información que podrán visualizar en cada módulo de la aplicación.
- Palabras claves: *Se definen en la etapa de Generalización.*
- Autor: Eduardo Cáceres
- Ventajas y desventajas.
  - Ventajas: Propone los lineamientos para modelar el acceso de cada uno de los usuarios del sistema en función de los requerimientos de la organización. Es de gran utilidad para sistemas de organizaciones complejas con muchos niveles jerárquicos.

## SOLUCIÓN PROPUESTA

- Desventajas: Solo especifica la estructura para definir cada uno de los perfiles de acceso del sistema. Posteriormente, el analista deberá elaborar cada uno de los perfiles siguiendo los lineamientos de este patrón.
- Condiciones y restricciones: Para que el patrón sea utilizado se necesita que el sistema tenga una composición compleja con varios niveles de acceso y privilegios de información.
- Elementos relacionados:
  - Diagrama conceptual del patrón (Figura 25).
  - Documento técnico de Especificación de Requisitos para el desarrollo del “Sistema Integrado de Información Universitaria (SIIU)” (SIIU, 2010).

Ahora que tenemos todos los atributos definidos, vamos a proceder a llenar la plantilla. El resultado que obtenemos es el que se puede apreciar en la Tabla 134.

Identificador:	PR0002	Clasificación:	Acceso al sistema
Nombre:	Perfil de usuario del sistema.		
Descripción:	Especifica los requisitos para definir los perfiles de acceso de los usuarios del software. Los perfiles limitan las acciones que podrán realizar los usuarios sobre el sistema informático y el grado de detalle de la información que podrán visualizar en cada módulo de la aplicación.		
Palabras claves:			
Autor:	Eduardo Cáceres		
Ventajas:	Propone los lineamientos para modelar el acceso de cada uno de los usuarios del sistema en función de los requerimientos de la organización. Es de gran utilidad para sistemas de organizaciones complejas con muchos niveles jerárquicos.		
Desventajas:	Solo especifica la estructura para definir cada uno de los perfiles de acceso del sistema. Posteriormente, el analista deberá elaborar cada uno de los perfiles siguiendo los lineamientos de este patrón.		
Condiciones:	Para que el patrón sea utilizado se necesita que el sistema tenga una composición compleja con varios niveles de acceso y privilegios de información.		

## SOLUCIÓN PROPUESTA

Requisitos																												
R01	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará el organismo al que pertenece y el nivel de actuación de su organización, ya sea a Universidad, Comunidad Autónoma, Ministerio de Educación y otras Instituciones.																											
R02	Los privilegios dados a un usuario, vendrán informados en el perfil que se asigne al usuario.																											
R03	Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios: Tipos de Usuario (R04) Niveles de Usuario (R05)																											
R04	<p>Para limitar el acceso a los apartados que se definan en la aplicación existirán cuatro tipos de usuario:</p> <p>Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general.</p> <p>Ejecución. Además de poseer los permisos del usuario de Lectura, podrá acceder a la aplicación para ejecutar y visualizar informes más especializados, previamente desarrollados por otro perfil de usuario. En este perfil estarán, por ejemplo, el Observatorio Universitario o la Fundación Universidad.es.</p> <p>Desarrollo. Además de poseer los permisos del usuario de Ejecución, podrá acceder a la aplicación para generar, ejecutar y visualizar informes. En este perfil estarán, por ejemplo, las Universidades, la ANECA, las CCAA.</p> <p>Administración. Además de poseer los permisos del usuario de Desarrollo, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado del Ministerio de Educación, de las CCAA y de las Universidades.</p>																											
R05	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz.</p> <table><tr><td></td><td>Detalle del dato (microdato)</td><td>Datos agregados</td><td>Indicadores</td></tr><tr><td>Área Académica</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área Económica</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área de Recursos Humanos</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área de Inserción Laboral</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área I+D</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr></table> <p>Dicha matriz deberá contemplar las distintas áreas (Académica, Económica, de Recursos Humanos, de Inserción Laboral e I+D) que existan en el sistema.</p>					Detalle del dato (microdato)	Datos agregados	Indicadores	Área Académica	Nivel	Nivel	Nivel	Área Económica	Nivel	Nivel	Nivel	Área de Recursos Humanos	Nivel	Nivel	Nivel	Área de Inserción Laboral	Nivel	Nivel	Nivel	Área I+D	Nivel	Nivel	Nivel
	Detalle del dato (microdato)	Datos agregados	Indicadores																									
Área Académica	Nivel	Nivel	Nivel																									
Área Económica	Nivel	Nivel	Nivel																									
Área de Recursos Humanos	Nivel	Nivel	Nivel																									
Área de Inserción Laboral	Nivel	Nivel	Nivel																									
Área I+D	Nivel	Nivel	Nivel																									

## SOLUCIÓN PROPUESTA

	<p>El nivel de usuario indicado en la matriz podrá tomar los siguientes valores:</p> <ul style="list-style-type: none"> <li>Nivel 4. No se tendrá acceso a este tipo de información.</li> <li>Nivel 3. Indica que el usuario sólo podrá acceder a los datos de la Universidad a la que esté asociado.</li> <li>Nivel 2. Indica que el usuario sólo podrá acceder a los datos de la Comunidad Autónoma a la que esté asociado, lo que implica que tendrá acceso a los datos de las Universidades que se localizan en dicha Comunidad Autónoma.</li> <li>Nivel 1. Corresponde a los usuarios que tienen el privilegio de acceder a todos los niveles de información.</li> </ul>
R06	Se asignarán tipo de usuario con sus respectivos niveles de seguridad a todos los organismos implicados: Ministerio de Educación, CCAA, Universidad, ANECA, CRUE, Observatorio Universitario de Becas y Ayudas al Estudio y Rendimiento Académico, Fundación Universidad.es, etc.
Elementos relacionados:	<ul style="list-style-type: none"> <li>Diagrama conceptual del patrón.</li> <li>Documento técnico de Especificación de Requisitos para el desarrollo de un “Sistema Integrado de Información Universitaria (SIIU)”.</li> </ul>

Tabla 134. Patrón de requisitos para una situación específica

Ahora que tenemos nuestro patrón, vamos a verificar que esta etapa esté completa mediante los *criterios de validación* respectivos:

- ¿Todos los campos de la plantilla tienen información? Sí.
- ¿Está claro el problema? Sí
- ¿Está clara la solución? Sí
- ¿El patrón resuelve completamente el problema? Sí

Finalmente, las *salidas* de esta etapa son: el *Patrón de requisitos para una situación específica* que podemos observar en la Tabla 134 y la *Documentación del patrón* que está enumerada en el atributo del patrón “Elementos relacionados”.

### 3.4.3.1.2 Control de calidad

Una vez que tuvimos el patrón estructurado en la plantilla propuesta para que pueda ser almacenado, es necesario revisarlo para garantizar su calidad. La entrada del *Control de calidad* se obtuvo en la etapa de *Identificación* (o Generalización, si fuera el caso) es el *Patrón de requisitos* (Tabla 134 (o Tabla 140)).

La principal *herramienta* de esta etapa es el *Modelo de calidad*, usaremos el modelo planteado anteriormente para evaluar la calidad de los patrones de requisitos.

Mediremos la calidad del patrón de requisitos a través de los siguientes factores de calidad definidos anteriormente: Atomicidad, Precisión, Abstracción, Inambigüedad, Comprensibilidad, Correctitud, Verificabilidad, Modificabilidad, Integridad, Frecuencia, Sencillez, Coherencia, Consistencia, Simplicidad y Adaptabilidad. En la Tabla 135, por

## SOLUCIÓN PROPUESTA

comodidad del lector, repetimos información de la Tabla 49, donde mostramos los factores con sus respectivos indicadores de medición.

		Atomicidad	Precisión	Abstracción	Inambigüedad	Comprensibilidad	Correctitud	Verificabilidad	Modificabilidad	Integridad	Frecuencia	Sencillez	Coherencia	Consistencia	Simplicidad	Adaptabilidad
Morfológicos	Número de elementos de la solución.	x							x	x					x	x
	Número de palabras.	x								x						x
	Promedio de palabras por oración.					x						x				
	Promedio de frases por párrafo.	x				x										x
	Número de abreviaturas o acrónimos.					x						x				
	Promedio de signos de puntuación por frase.				x	x										
Léxicos	Número de términos conectivos.	x	x		x	x				x						
	Número de términos imprecisos.		x		x			x								
	Número de términos de diseño.			x					x			x				x
Analíticos	Número de errores de ortografía y gramática.				x	x	x					x				
	Número de términos ambiguos.		x		x	x										
	Número de formas verbales condicionales y compuestas.		x									x				
	Número de términos de dominio.		x	x								x				x
Relacionales	Número de relaciones con otros elementos.	x							x							x
	Número de dependencias entre elementos del activo.								x					x	x	x

Tabla 135. Relación entre los factores de calidad y los indicadores de calidad (obtenida de la Tabla 49)

Aplicando las técnicas: *Inspección* y *Juicio de expertos* analizamos el patrón y obtenido para cada uno de los indicadores el resultado que mostramos en la Tabla 136. Considerando las calificaciones ‘Bueno’, ‘Regular’ y ‘Malo’, que equivalen a los puntajes 1, 2 y 3 respectivamente.

## SOLUCIÓN PROPUESTA

		Atomicidad	Precisión	Abstracción	Inambigüedad	Comprensibilidad	Correctitud	Verificabilidad	Modificabilidad	Integridad	Frecuencia	Sencillez	Coherencia	Consistencia	Simplicidad	Adaptabilidad
Morfológicos	Número de elementos de la solución.	2							2	2					2	2
	Número de palabras.	1								2						2
	Promedio de palabras por oración.					2						3				
	Promedio de frases por párrafo.	3				3										1
	Número de abreviaturas o acrónimos.					1						1				
	Promedio de signos de puntuación por frase.				3	3										
Léxicos	Número de términos conectivos.	1	1		1	1				3						
	Número de términos imprecisos.		1		1			1								
	Número de términos de diseño.			1					1			1				1
Analíticos	Número de errores de ortografía y gramática.				1	1	1					1				
	Número de términos ambiguos.		1		1	1										
	Número de formas verbales condicionales y compuestas.		1									1				
	Número de términos de dominio.		1	1								1				1
Relacionales	Número de relaciones con otros elementos.	1							1							1
	Número de dependencias entre elementos del activo.								1					3	1	1

*Tabla 136. Resultados de cada indicador.*

Lo siguiente que debemos hacer es calcular los ponderados según los pesos establecidos en el apartado 3.3 Modelo de Calidad. La Tabla 137 nos muestra los resultados promedio de cada factor analizado.

## SOLUCIÓN PROPUESTA

	Resultado por factor
Atomicidad	1.33
Precisión	1
Abstracción	1
Inambigüedad	1.4
Comprensibilidad	1.8
Correctitud	1
Verificabilidad	1
Modificabilidad	1.17
Integridad	2.5
Sencillez	1.35
Consistencia	3
Simplicidad	1.5
Adaptabilidad	1.28

Tabla 137. Resultado de cada factor.

Para finalizar, vamos a verificar los *criterios de validación*:

- ¿Se ha aplicado correcta y objetivamente el modelo de calidad? Sí, el modelo se ha aplicado correctamente.
- ¿El patrón cumple los requisitos mínimos de calidad exigidos por la organización? Sí, el resultado del Control de calidad en la mayoría de factores ha sido “Bueno” salvo en Integridad y Consistencia en los que el resultado fue “Malo”

En esta etapa, las salidas son los resultados de las evaluaciones del patrón y las observaciones a la calidad del patrón.

La principal observación que podemos hacer a este patrón es que tiene una puntuación baja para el factor de consistencia, esto fue causado principalmente por solapamientos entre sus elementos. A pesar de esta observación, consideramos que el patrón tiene la calidad suficiente para no ser descartado, pero es necesario pasar por proceso de *Generalización* para mejorarlo.

### 3.4.3.1.3 Generalización del activo

En esta etapa de *Generalización*, las entradas que utilizaremos las hemos obtenido en las etapas anteriores (*Identificación y Control de calidad*). El patrón de requisitos para una situación específica, la documentación del patrón y las observaciones a la calidad del patrón.

Siguiendo lo que indica la etapa de *Generalización*, debemos seguir cuatro pasos para generalizar nuestro patrón. La aplicación de los dos primeros pasos lo podemos observar en la Tabla 138 de la siguiente manera:

- **Análisis.** Utilizamos la técnica de *Abstracción*. Identificamos la información de “relleno”, es decir, que no es relevante para nuestro patrón. Esta información la tachamos (~~tachamos~~) en señal de que debe ser borrada del patrón.



## SOLUCIÓN PROPUESTA

- Clasificación. Utilizamos la técnica de *Generalización* para distinguir la información circunstancial del patrón. Esta información es la que consideramos que va a variar en cada uno de los sistemas en los que un analista reúse nuestro patrón y la subrayamos (subrayamos) para reemplazarla posteriormente por variables.

Identificador:	PR0002	Clasificación:	Acceso al sistema
Nombre:	Perfil de usuario del sistema		
Descripción:	Especifica los requisitos para definir los perfiles de acceso de los usuarios del software. Los perfiles limitan las acciones que podrán realizar los usuarios sobre el sistema informático y el grado de detalle de la información que podrán visualizar en cada módulo de la aplicación.		
Palabras claves:			
Autor:	Eduardo Cáceres – Universidad de Piura		
Ventajas:	Propone los lineamientos para modelar el acceso de cada uno de los usuarios del sistema en función de los requerimientos de la organización. Es de gran utilidad para sistemas de organizaciones complejas con muchos niveles jerárquicos.		
Desventajas:	Solo especifica la estructura para definir cada uno de los perfiles de acceso del sistema. Posteriormente, el analista deberá elaborar cada uno de los perfiles siguiendo los lineamientos de este patrón.		
Condiciones:	Para que el patrón sea utilizado se necesita que el sistema tenga una composición compleja con varios niveles de acceso y privilegios de información.		
Requisitos			
R01	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará <u>el organismo</u> al que pertenece y el <u>nivel de actuación</u> de su organización, <del>ya sea a Universidad, Comunidad Autónoma, Ministerio de Educación y otras Instituciones.</del>		
R02	Los privilegios dados a un usuario, vendrán informados en el perfil que se asigne al usuario.		
R03	Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios: Tipos de Usuario (R04) Niveles de Usuario (R05)		
R04	Para limitar el acceso a los apartados que se definan en la aplicación existirán <u>cuatro</u> tipos de usuario: Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general. Ejecución. Además de poseer los permisos del usuario de Lectura, podrá		

## SOLUCIÓN PROPUESTA

	<p>acceder a la aplicación para ejecutar y visualizar informes más especializados, previamente desarrollados por otro perfil de usuario. <del>En este perfil estarán, por ejemplo, el Observatorio Universitario o la Fundación Universidad.es.</del> Desarrollo. Además de poseer los permisos del usuario de Ejecución, podrá acceder a la aplicación para generar, ejecutar y visualizar informes. <del>En este perfil estarán, por ejemplo, las Universidades, la ANECA, las CCAA.</del> Administración. Además de poseer los permisos del usuario de Desarrollo, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado <del>del Ministerio de Educación, de las CCAA y de las Universidades.</del></p>																								
R05	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz.</p> <table><tr><td></td><td>Detalle del dato (microdato)</td><td>Datos agregados</td><td>Indicadores</td></tr><tr><td>Área Académica</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área Económica</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área de Recursos Humanos</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área de Inserción Laboral</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>Área I+D</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr></table> <p>Dicha matriz deberá contemplar las distintas áreas (Académica, Económica, de Recursos Humanos, de Inserción Laboral e I+D) que existan en el sistema. El nivel de usuario indicado en la matriz podrá tomar los siguientes valores: Nivel 4. No se tendrá acceso a este tipo de información. Nivel 3. Indica que el usuario sólo podrá acceder a los datos de la Universidad a la que esté asociado. Nivel 2. Indica que el usuario sólo podrá acceder a los datos de la Comunidad Autónoma a la que esté asociado, lo que implica que tendrá acceso a los datos de las Universidades que se localizan en dicha Comunidad Autónoma. Nivel 1. Corresponde a los usuarios que tienen el privilegio de acceder a todos los niveles de información.</p>		Detalle del dato (microdato)	Datos agregados	Indicadores	Área Académica	Nivel	Nivel	Nivel	Área Económica	Nivel	Nivel	Nivel	Área de Recursos Humanos	Nivel	Nivel	Nivel	Área de Inserción Laboral	Nivel	Nivel	Nivel	Área I+D	Nivel	Nivel	Nivel
	Detalle del dato (microdato)	Datos agregados	Indicadores																						
Área Académica	Nivel	Nivel	Nivel																						
Área Económica	Nivel	Nivel	Nivel																						
Área de Recursos Humanos	Nivel	Nivel	Nivel																						
Área de Inserción Laboral	Nivel	Nivel	Nivel																						
Área I+D	Nivel	Nivel	Nivel																						
R06	<p>Se asignarán tipo de usuario con sus respectivos niveles de seguridad a todos <u>los organismos</u> implicados: Ministerio de Educación, CCAA, Universidad, ANECA, CRUE, Observatorio Universitario de Becas y Ayudas al Estudio y Rendimiento Académico, Fundación Universidad.es, etc.</p>																								
Elementos relacionados:	<p>Diagrama conceptual del patrón. Documento técnico de Especificación de Requisitos para el desarrollo de un “Sistema Integrado de Información Universitaria (SIIU)”.</p>																								

*Tabla 138. Análisis y clasificación de la información del patrón.*

Con el patrón analizado y clasificado, debemos proceder con los siguientes pasos de esta etapa:

## SOLUCIÓN PROPUESTA

- Parametrización. Trabajamos la información que estaba subrayada en el patrón de la Tabla 138 para que los enunciados sean generales. En algunos casos solamente fue necesario modificar la redacción para lograr el propósito. Sin embargo, en otros tuvimos que reemplazar las palabras por variables y en simultáneo elaborar el *Diccionario del patrón de requisitos* que podemos observar en la Tabla 139.

N°	Variable	Nombre	Descripción	Ejemplo
1	<N_Tipos>	Cantidad de tipos de usuario	Número de tipos de usuario que tendrá el sistema.	cuatro
2	<Tipo_usuario2> ... <Tipo_usuarioN-1>	Tipo de usuario	Cada tipo describe las acciones que puede realizar el usuario cuando accede al sistema. Se deben ordenar según su ámbito de acción de menor a mayor. Se han establecido dos por defecto: Lectura y Administración.	Ejecución. Además de poseer los permisos del usuario de Lectura, podrá acceder a la aplicación para ejecutar y visualizar informes más especializados, previamente desarrollados por otro perfil de usuario.
3	<Tipo_dato1> ... <Tipo_datoN>	Tipo de dato	Son los tipos de dato que maneja el sistema. Se recomienda ordenarlos del más detallado al más general.	Datos de detalle Datos agregados Indicadores
4	<Área_1> ... <Área_M>	Área	Son las áreas o módulos de información que tiene el sistema. Está relacionado directamente con las áreas de la organización.	Área Académica Área de RRHH Área de Marketing
5	<Nivel_N-2> ... <Nivel_2>	Nivel de acceso	Define la información de las organizaciones a las que tiene acceso el usuario. Se deben ordenar según su jerarquía de menor a mayor.	Nivel 2. Indica que el usuario sólo podrá acceder a los datos de las organizaciones con igual nivel jerárquico a la organización que esté asociado.

Tabla 139. Diccionario del patrón de requisitos.

- Revisión. Finalmente revisamos el patrón y mejoramos su contenido, redacción y forma. El resultado fue el *Patrón de requisitos* de la Tabla 140.

## SOLUCIÓN PROPUESTA

Identificador:	PR0002	Clasificación:	Acceso al sistema
Nombre:	Perfil de usuario del sistema		
Descripción:	Especifica los requisitos para definir los perfiles de acceso de los usuarios del software. Los perfiles limitan las acciones que podrán realizar los usuarios sobre el sistema informático y el grado de detalle de la información que podrán visualizar en cada módulo de la aplicación.		
Palabras claves:			
Autor:	Eduardo Cáceres – Universidad de Piura		
Ventajas:	Propone los lineamientos para modelar el acceso de cada uno de los usuarios del sistema en función de los requerimientos de la organización. Es de gran utilidad para sistemas de organizaciones complejas con muchos niveles jerárquicos.		
Desventajas:	Solo especifica la estructura para definir cada uno de los perfiles de acceso del sistema. Posteriormente, el analista deberá elaborar cada uno de los perfiles siguiendo los lineamientos de este patrón.		
Condiciones:	Para que el patrón sea utilizado se necesita que el sistema tenga una composición compleja con varios niveles de acceso y privilegios de información.		
Requisitos			
R01	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará: el nombre de la organización a la que pertenece y el tipo de organización.		
R02	Los privilegios dados a un usuario, vendrán informados en el perfil que se asigne al usuario.		
R03	Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios: Tipos de Usuario (R04) Niveles de Usuario (R05)		
R04	Para limitar el acceso a los módulos que se definan en la aplicación existirán <N_Tipos> tipos de usuario: Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general. <Tipo_usuario2> ..... <Tipo_usuarioN-1> Administración. Además de poseer los permisos del usuario <Tipo_N-1>, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado de TI.		
R05	Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz:		

## SOLUCIÓN PROPUESTA

	<table><tr><td></td><td>&lt;Tipo_dato1&gt;</td><td>.....</td><td>&lt;Tipo_datoN&gt;</td></tr><tr><td>&lt;Área_1&gt;</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>.....</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>&lt;Área_M&gt;</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr></table> <p>Dicha matriz deberá contemplar las distintas áreas que existan en el sistema. El nivel de usuario indicado en la matriz podrá tomar los siguientes valores: Nivel N. No se tendrá acceso a este tipo de información. Nivel N-1. Indica que el usuario sólo podrá acceder a los datos de la organización a la que esté asociado. &lt;Nivel_N-2&gt; ..... &lt;Nivel_2&gt; Nivel 1. Corresponde a los usuarios que tienen el privilegio de acceder a los datos de todas las organizaciones del sistema.</p>		<Tipo_dato1>	.....	<Tipo_datoN>	<Área_1>	Nivel	Nivel	Nivel	.....	Nivel	Nivel	Nivel	<Área_M>	Nivel	Nivel	Nivel
	<Tipo_dato1>	.....	<Tipo_datoN>														
<Área_1>	Nivel	Nivel	Nivel														
.....	Nivel	Nivel	Nivel														
<Área_M>	Nivel	Nivel	Nivel														
R06	Se asignarán tipos de usuario con sus respectivos niveles de seguridad a todas las organizaciones implicadas en el sistema.																
Elementos relacionados:	Diagrama conceptual del patrón. Documento técnico de Especificación de Requisitos para el desarrollo de un “Sistema Integrado de Información Universitaria (SIIU)”.																

*Tabla 140. Patrón de requisitos.*

Para finalizar la etapa de *Generalización*, verificamos los *criterios de validación*:

- ¿Todos los campos del patrón han sido generalizados? / ¿Existe todavía información específica en algún elemento de la plantilla? Sí, el patrón de la Tabla 140 es general y puede ser reutilizado en otros proyectos.
- ¿El contenido del patrón es accesible al entendimiento de cualquier persona? Sí, el patrón está escrito en un lenguaje sencillo para que pueda ser entendido por una persona con los mínimos conocimientos de Ingeniería de Sistemas y de Requisitos.

De esta manera, las salidas son el *Patrón de requisitos* (Tabla 140) y el *Diccionario del patrón de requisitos* (Tabla 139) obtenidos anteriormente.

### 3.4.3.1.4 Indización

En este punto del proceso, ya tenemos un *Patrón de requisitos* que ha pasado un control de calidad y está listo para almacenarse en el repositorio, este patrón es la entrada en esta etapa. Para esto, necesitamos extraer los términos de indización de su contenido y adaptarlo a un modelo de representación de la información.

## SOLUCIÓN PROPUESTA

---

En esta etapa del proceso hemos identificado herramientas que nos van a ayudar a extraer del patrón los términos de indización. La primera herramienta es el *Modelo de representación de la información* que en este caso como no contamos con una herramienta informática vamos a utilizar la indización manual.

Con el modelo definido, vamos a aplicar la *indización* propiamente dicha analizando el contenido del patrón y extrayendo descriptores simples y compuestos. El resultado del análisis es el siguiente:

- Perfil
- Acceso
- Privilegio de información
- Tipo
- Nivel
- Usuario

Una vez que hemos determinado los descriptores para nuestro patrón, vamos a revisar los *criterios de validación*:

- ¿Se han determinado los términos de indización más adecuados para el documento? Sí, a simple vista se puede afirmar que los descriptores obtenidos resumen el contenido del patrón de requisitos.

Finalmente, nuestra salida en la etapa de *Indización* fue la *Información para la indización del patrón* que enunciamos a continuación y que debe ser agregada en la plantilla del patrón de requisitos PR0002, en el atributo *Palabras Claves*.

- Palabras claves: Perfil, Acceso, Privilegio de información, Tipo, Nivel y Usuario.

### **3.4.3.1.5 Almacenamiento**

Finalmente tenemos que almacenar el patrón. Las entradas de esta etapa son el *Patrón de requisitos (incluyendo los elementos relacionados, principalmente el documento original de requisitos, importante para la comprensión del patrón generalizado)*, obtenido en el paso de *Generalización*; la *Información para la indización del patrón*, obtenida en la etapa de *Indización*; y el *Catálogo de patrones*.

Como se mencionó al inicio de esta sección el ejemplo se realizó de manera manual; sin embargo, resaltaremos que utilizar una *Herramienta informática* para el almacenamiento es lo más apropiado. En su lugar, la *técnica* que utilizamos para colocarlo en un catálogo físico es el *Archivamiento*. Para efectos del ejemplo, solo explicaremos lo que se debe hacer:

- Clasificar el patrón según su contenido para determinar su ubicación dentro del catálogo.
- Ordenar cada uno de sus componentes en la sección del catálogo que corresponda.
- Instalar los documentos, físicos o digitales, en el catálogo y guardarlo.

## SOLUCIÓN PROPUESTA

---

Luego de seguir estos tres pasos, verificamos los *criterios de validación*:

- ¿El patrón ha sido incorporado al repositorio?
- ¿El patrón estará disponible para futuras consultas?

Si el proceso se realizó correctamente, se puede asegurar que la respuesta a ambos criterios es afirmativa. De esta manera, nuestra salida sería un nuevo patrón de requisitos en el repositorio, es decir, el *Catálogo de patrones actualizado*.

### 3.4.3.2 Recuperación

Para ejemplificar el proceso de *Recuperación*, vamos a suponer que estamos trabajando en un proyecto de software para el Grupo Empresarial CS<sup>12</sup>.

En primer lugar, vamos a hacer una breve introducción de la organización. El Grupo CS es un conglomerado industrial de capitales peruanos con negocios presentes en Perú, como también en Bolivia, Colombia, Ecuador, Argentina y Puerto Rico. Sus actividades se desarrollan en los sectores de alimentos, cemento, papeles, agroindustria, transporte y servicios; todos ellos focalizados en la calidad del producto o servicio que se entrega al consumidor en todo momento.

Para efectos de este ejemplo, vamos a suponer que la estructura organizacional del grupo está dividido en cuatro niveles: Grupo, Rubro, Empresa y Unidad de Negocio (sucursales, plantas industriales, locales comerciales, etc.); los cuales están ilustrados en la Figura 26.

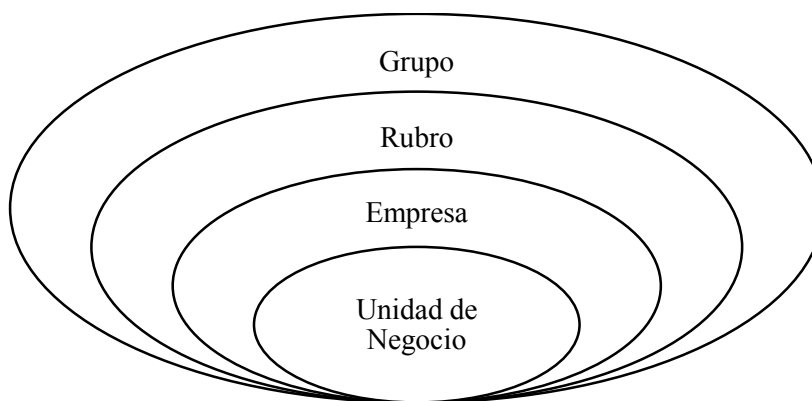


Figura 26. Estructura organizacional del Grupo CS

A continuación, en la Figura 27 podemos visualizar algunos de los elementos que están comprendidos en cada nivel de la estructura.

---

<sup>12</sup> El nombre “Grupo Empresarial CS” o “Grupo CS” es ficticio y ha sido utilizado con fines didácticos.

## SOLUCIÓN PROPUESTA

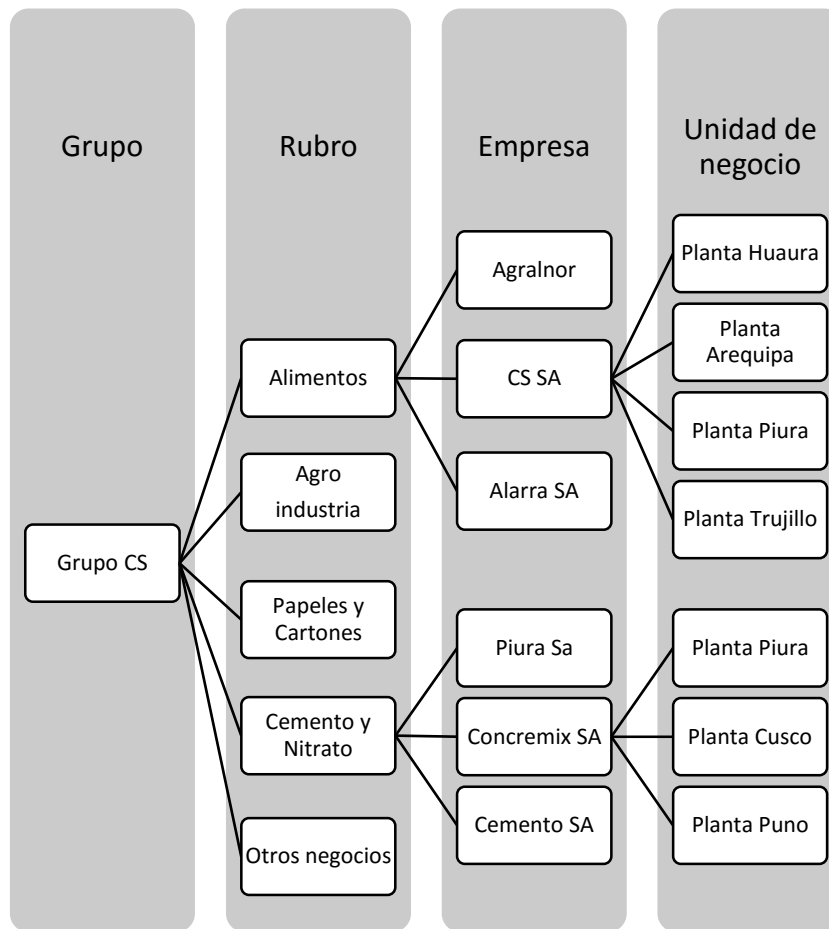


Figura 27. Estructura de las empresas del Grupo CS

En segundo lugar, explicamos el objetivo del proyecto: “Desarrollar un sistema informático que gestione y consolide la información operacional, financiera y comercial de todas las empresas del grupo a través de reportes de performance y estado que sirvan para el monitoreo y la toma de decisiones de sus directivos”.

Suponemos que estamos trabajando en el proyecto y nos encontramos en la etapa de análisis. Debido a que la organización donde trabajamos cuenta con un catálogo de patrones de requisitos, hemos decidido utilizarlo para simplificar nuestro trabajo y garantizar la calidad de los requisitos elicitados.

### 3.4.3.2.1 Consulta

Según la metodología propuesta, necesitamos tres entradas para realizar la consulta.

En primer lugar, tenemos el *Catálogo de patrones* de la organización para la cual trabajamos.

En segundo lugar, vamos a describir la necesidad en el *Enunciado del problema*, el cual quedaría redactado de la siguiente manera: “Necesitamos definir los permisos de cada uno de los actores en el sistema y determinar la información a la que puede acceder cada persona debido a que el sistema va a gestionar información de las diversas empresas



## SOLUCIÓN PROPUESTA

del Grupo CS. Dependiendo de sus funciones, cada usuario debe tener permisos para visualizar y/o modificar únicamente la información que le compete”.

La última entrada es la *Consulta* propiamente dicha que es una síntesis del enunciado anterior. Para este ejemplo, nuestra consulta podría ser: “Permisos de usuario y acceso a la información”.

El siguiente paso consiste en utilizar una *herramienta* para localizar patrones de requisitos que puedan ser una solución a nuestro problema. Lo más conveniente es que utilicemos una *Herramienta informática* de búsqueda. Si el catálogo no está automatizado, debemos recurrir a un *Índice de búsqueda* manual.

Finalmente, la salida de esta etapa del proceso es el conjunto de resultados que ha tenido nuestra consulta. Para este ejemplo en específico, suponemos que obtuvimos solo dos *Resultados de búsqueda* listados en la Tabla 141.

Identificador	Clasificación	Nombre	Descripción
PR0001	Seguridad	Nivel 2 de acceso a un sistema	Especifica los requisitos para definir la seguridad en el acceso a un sistema de información. Este grado de acceso debe considerar las restricciones del nivel 2 según la clasificación de dificultad de la empresa Asynconyx.
PR0002	Acceso al sistema	Perfil de usuario del sistema	Especifica los requisitos para definir los perfiles de acceso de los usuarios del software. Los perfiles limitan las acciones que podrán realizar los usuarios sobre el sistema informático y el grado de detalle de la información que podrán visualizar en cada módulo de la aplicación.

Tabla 141. Resultados de la búsqueda.

Para confirmar que los resultados van a ser útiles para nuestro proyecto, debemos revisar los *criterios de validación*:

- ¿Se han obtenido suficientes resultados de búsqueda? En este caso se obtuvieron dos resultados, pero son suficientes para el propósito de nuestra consulta.
- A priori, ¿Los resultados satisfacen las necesidades que originaron la consulta? Después de una revisión superficial de los resultados podemos intuir que los patrones obtenidos son lo que estamos buscando.

### 3.4.3.2.2 Selección

En esta etapa de Selección, tenemos como primera entrada el *Resumen de los patrones encontrados*, es decir, del resultado de la búsqueda. Podemos revisar solamente la información de la Tabla 141 o si creemos conveniente, acceder a mayor información del patrón y revisar sus atributos “Ventajas y desventajas” o “Condiciones”.

Adicionalmente, tenemos como entradas el *Proyecto actual* y el *Enunciado del problema*, que se definió en la etapa de *Consulta*. Ambos elementos describen lo que estamos buscando y determinan el objetivo de nuestra búsqueda.

## SOLUCIÓN PROPUESTA

---

Ahora que disponemos de información sobre las alternativas del patrón y conocemos qué es lo que necesitamos, utilizamos la *herramienta Evaluación y comparación* para analizar los dos patrones que obtuvimos en la consulta y escoger el que más nos conviene. Esta evaluación es complementada con el *Juicio de expertos* que está basado en las experiencias pasadas.

En esta etapa la selección es similar a la que realizamos cuando tenemos la lista de resultados en un buscador web y decidimos que páginas abrir y cuáles no, basándonos en un breve resumen del contenido. En nuestro caso podemos identificar los mejores patrones y abrirlos para revisarlos por completo y tomar una mejor decisión.

Luego de revisar los dos resultados de nuestra consulta, explicaremos los motivos que nos llevan a seleccionar el patrón PR0002:

- El patrón PR0001 solo define aspectos de acceso al sistema en cuanto a usuario y contraseña, en contraste con el patrón PR0002 que define perfiles y niveles de acceso a la información.
- El PR0002 considera una estructura con diversos niveles de acción, mientras que el PR0001 considera a todos los usuarios por igual.
- El PR0002 estructura la información de una manera ordenada que va a facilitar la documentación del sistema.

Por estos motivos decidimos utilizar el Patrón de requisitos PR0002 y para integrarlo al proyecto, es necesario extraigamos por completo del repositorio. Para esto, podemos apoyarnos de una herramienta informática o hacerlo de forma manual. El resultado de esa recuperación está compuesto principalmente por tres elementos:

- El patrón de requisitos (Tabla 140)
- El diccionario del patrón (Tabla 139)
- Elementos Relacionados. Documentos complementarios al patrón. En este caso específico son el Diagrama conceptual del patrón y el Documento técnico de Especificación de Requisitos para el desarrollo de un “Sistema Integrado de Información Universitaria (SIIU)”.

Con el patrón recuperado por completo, podemos contestar los *criterios de validación*:

- ¿Los patrones seleccionados son compatibles con el proyecto? En este caso se ha elegido solo un patrón. Después de analizarlo se ha llegado a la conclusión de que se puede integrar sin mayor dificultad al proyecto.
- ¿Los patrones seleccionados son suficientes? Sí, el patrón cubre todos los puntos del enunciado del problema en su totalidad.

La salida de esta etapa es el *Patrón de requisitos seleccionado* que está compuesto por los tres elementos antes mencionados. Para efectos del ejemplo, vamos a colocar nuevamente el primer y principal elemento: El patrón de requisitos (Tabla 142).

## SOLUCIÓN PROPUESTA

Identificador:	PR0002	Clasificación:	Acceso al sistema
Nombre:	Perfil de usuario del sistema		
Descripción:	Especifica los requisitos para definir los perfiles de acceso de los usuarios del software. Los perfiles limitan las acciones que podrán realizar los usuarios sobre el sistema informático y el grado de detalle de la información que podrán visualizar en cada módulo de la aplicación.		
Palabras claves:	Perfil de acceso, tipo de usuario, privilegio		
Autor:	Eduardo Cáceres – Universidad de Piura		
Ventajas:	<ul style="list-style-type: none"><li>• Propone los lineamientos para modelar el acceso de cada uno de los usuarios del sistema en función de los requerimientos de la organización.</li><li>• Es de gran utilidad para sistemas de organizaciones complejas con muchos niveles jerárquicos.</li></ul>		
Desventajas:	<ul style="list-style-type: none"><li>• Solo especifica la estructura para definir cada uno de los perfiles de acceso del sistema. Posteriormente, el analista deberá elaborar cada uno de los perfiles siguiendo los lineamientos de este patrón.</li></ul>		
Condiciones:	Para que el patrón sea utilizado se necesita que el sistema tenga una composición compleja con varios niveles de acceso y privilegios de información.		
Requisitos			
R01	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará: el nombre de la organización a la que pertenece y el tipo de organización.		
R02	Los privilegios dados a un usuario, vendrán informados en el perfil que se asigne al usuario.		
R03	Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios: <ul style="list-style-type: none"><li>• Tipos de Usuario (R04)</li><li>• Niveles de Usuario (R05)</li></ul>		
R04	Para limitar el acceso a los módulos que se definan en la aplicación existirán <N_Tipos> tipos de usuario: <ul style="list-style-type: none"><li>• Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general.</li><li>• &lt;Tipo_usuario2&gt;</li><li>• .....</li><li>• &lt;Tipo_usuarioN-1&gt;</li><li>• Administración. Además de poseer los permisos del usuario &lt;Tipo_N-1&gt;, tendrá acceso a la parte de administración (seguridad, parametrización...)</li></ul>		

## SOLUCIÓN PROPUESTA

	del Sistema. En este perfil estará el personal especializado de TI.																
R05	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz:</p> <table><tr><td></td><td>&lt;Tipo_dato1&gt;</td><td>.....</td><td>&lt;Tipo_datoN&gt;</td></tr><tr><td>&lt;Área_1&gt;</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>.....</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr><tr><td>&lt;Área_M&gt;</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr></table> <p>Dicha matriz deberá contemplar las distintas áreas que existan en el sistema. El nivel de usuario indicado en la matriz podrá tomar los siguientes valores:</p> <ul style="list-style-type: none"><li>• Nivel N. No se tendrá acceso a este tipo de información.</li><li>• Nivel N-1. Indica que el usuario sólo podrá acceder a los datos de la organización a la que esté asociado.</li><li>• &lt;Nivel_N-2&gt;</li><li>• .....</li><li>• &lt;Nivel_2&gt;</li><li>• Nivel 1. Corresponde a los usuarios que tienen el privilegio de acceder a los datos de todas las organizaciones del sistema.</li></ul>		<Tipo_dato1>	.....	<Tipo_datoN>	<Área_1>	Nivel	Nivel	Nivel	.....	Nivel	Nivel	Nivel	<Área_M>	Nivel	Nivel	Nivel
	<Tipo_dato1>	.....	<Tipo_datoN>														
<Área_1>	Nivel	Nivel	Nivel														
.....	Nivel	Nivel	Nivel														
<Área_M>	Nivel	Nivel	Nivel														
R06	Se asignarán tipos de usuario con sus respectivos niveles de seguridad a todas las organizaciones implicadas en el sistema.																
Elementos relacionados:	<ul style="list-style-type: none"><li>• Diagrama conceptual del patrón.</li><li>• Documento técnico de Especificación de Requisitos para el desarrollo de un “Sistema Integrado de Información Universitaria (SIIU)”.</li></ul>																

*Tabla 142. El patrón de requisitos (obtenido de la Tabla 140)*

### 3.4.3.2.3 Integración

En esta etapa, la primera entrada es *el Patrón de requisitos seleccionado* obtenido en la etapa anterior. A pesar de que el patrón está conformado por tres elementos, solo vamos a integrar en el proyecto el Patrón propiamente dicho incluyendo sus elementos relacionados. El resto de información sirve como soporte para ayudarnos a integrarlo.

La segunda entrada es el *Proyecto actual* y está muy relacionada con la herramienta *Plantillas y formatos* que consiste en los lineamientos de forma que estamos utilizando para desarrollar el Documento de Especificación de Requisitos del proyecto de

## SOLUCIÓN PROPUESTA

software para el Grupo CS. Para nuestro ejemplo, el formato que estamos utilizando para los requisitos es el que se puede observar en la Tabla 143.

Identificador:		Versión:	
Tipo:		Autor:	
Prioridad:		Necesidad:	
Descripción:			
Requisitos Relacionados:			

*Tabla 143. Plantilla de requisitos del Proyecto de Software para el Grupo CS.*

El paso siguiente para la integración sería trasladar la información del Patrón PR0002 al formato de la Tabla 143 que corresponde a nuestro proyecto actual. A simple vista, nos damos cuenta que las estructuras son distintas y tendremos que buscar correspondencia entre los elementos de ambas.

En las tablas a continuación (Tabla 144, Tabla 149) podemos observar el patrón de requisitos que hemos seleccionado integrado a las plantillas del proyecto actual.

Identificador:		Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:		Necesidad:	
Descripción:	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará: el nombre de la organización a la que pertenece y el tipo de organización.		
Requisitos Relacionados:			

*Tabla 144. Requisito R01 integrado*

Identificador:		Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:		Necesidad:	
Descripción:	Los privilegios dados a un usuario, vendrán informados en el perfil que se asigne al usuario.		
Requisitos Relacionados:			

*Tabla 145. Requisito R02 integrado*

## SOLUCIÓN PROPUESTA

Identificador:		Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:		Necesidad:	
Descripción:	Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios: Tipos de Usuario (R04) Niveles de Usuario (R05)		
Requisitos Relacionados:			

*Tabla 146. Requisito R03 integrado*

Identificador:		Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:		Necesidad:	
Descripción:	<p>Para limitar el acceso a los módulos que se definan en la aplicación existirán &lt;N_Tipos&gt; tipos de usuario:</p> <ul style="list-style-type: none"> <li>• Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general.</li> <li>• &lt;Tipo_usuario2&gt;</li> <li>• .....</li> <li>• &lt;Tipo_usuarioN-1&gt;</li> <li>• Administración. Además de poseer los permisos del usuario &lt;Tipo_N-1&gt;, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado de TI.</li> </ul>		
Requisitos Relacionados:			

*Tabla 147. Requisito R04 integrado*

Identificador:		Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:		Necesidad:	
Descripción:	Se asignarán tipos de usuario con sus respectivos niveles de seguridad a todas las organizaciones implicadas en el sistema.		
Requisitos Relacionados:			

*Tabla 148. Requisito R05 integrado*

## SOLUCIÓN PROPUESTA

Identificador:		Versión:	01																
Tipo:	Acceso al sistema	Autor:	E. Cáceres																
Prioridad:		Necesidad:																	
Descripción:	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz:</p> <table border="1"> <tr> <td></td><td>&lt;Tipo_dato1&gt;</td><td>.....</td><td>&lt;Tipo_datoN&gt;</td></tr> <tr> <td>&lt;Área_1&gt;</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>.....</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>&lt;Área_M&gt;</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> </table> <p>Dicha matriz deberá contemplar las distintas áreas que existan en el sistema.  El nivel de usuario indicado en la matriz podrá tomar los siguientes valores:  Nivel N. No se tendrá acceso a este tipo de información.  Nivel N-1. Indica que el usuario sólo podrá acceder a los datos de la organización a la que esté asociado.  &lt;Nivel_N-2&gt;  .....  &lt;Nivel_2&gt;  Nivel 1. Corresponde a los usuarios que tienen el privilegio de acceder a los datos de todas las organizaciones del sistema.</p>				<Tipo_dato1>	.....	<Tipo_datoN>	<Área_1>	Nivel	Nivel	Nivel	.....	Nivel	Nivel	Nivel	<Área_M>	Nivel	Nivel	Nivel
	<Tipo_dato1>	.....	<Tipo_datoN>																
<Área_1>	Nivel	Nivel	Nivel																
.....	Nivel	Nivel	Nivel																
<Área_M>	Nivel	Nivel	Nivel																
Requisitos Relacionados:																			

*Tabla 149. Requisito R06 integrado*

Una vez que tenemos cada requisito en su respectiva plantilla, el siguiente paso es colocar las plantillas en el documento de especificación de requisitos del proyecto. Luego debemos verificar el *criterio de validación*:

- ¿El patrón ha sido integrado al proyecto? Sí.

En nuestro caso, como no tenemos el documento de especificación completo seguiremos trabajando con los requisitos en sus respectivas plantillas. Estos seis requisitos son la salida de esta etapa del proceso.

### 3.4.3.2.4 Adecuación

Siguiendo con los requisitos obtenidos en la etapa anterior, la principal entrada para esta etapa es el *patrón integrado al proyecto*, que en este caso está formado por los seis requisitos en sus respectivas plantillas.

La *herramienta* que vamos a utilizar se llama *parametrización*, vamos a convertir este patrón general en un conjunto de requisitos específicos para este proyecto. En nuestro

## SOLUCIÓN PROPUESTA

ejemplo, los requisitos que requieren ser instanciados son el R04 (Tabla 147) y el R05 (Tabla 148); los cuales quedan según la Tabla 150 y la Tabla 151, respectivamente.

Identificador:		Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:		Necesidad:	
Descripción:	<p>Para limitar el acceso a los módulos que se definan en la aplicación existirán 5 tipos de usuario:</p> <ul style="list-style-type: none"> <li>• Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general.</li> <li>• Escritura. Además de poder visualizar los informes predefinidos, podrá acceder al software para ingresar o cargar información en el sistema. Este perfil se utilizará para almacenar nueva información en la base de datos.</li> <li>• Ejecución. Además de poder acceder a la aplicación para visualizar los informes previamente ejecutados, podrá ejecutar y personalizar cada uno de los reportes previamente diseñados por otros usuarios.</li> <li>• Diseño. Además de poder realizar las acciones de los usuarios de tipo Lectura y Ejecución, podrá acceder a la aplicación para crear nuevos reportes para la aplicación.</li> <li>• Administración. Además de poseer los permisos del usuario de Escritura y el de Diseño, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado de TI.</li> </ul>		
Requisitos Relacionados:			

*Tabla 150. Requisito R04 instanciado*



## SOLUCIÓN PROPUESTA

Identificador:		Versión:	01																																				
Tipo:	Acceso al sistema	Autor:	E. Cáceres																																				
Prioridad:		Necesidad:																																					
Descripción:	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz:</p> <table border="1"> <thead> <tr> <th></th><th>Datos detallados</th><th>Datos consolidados</th><th>KPIs</th></tr> </thead> <tbody> <tr> <td>Administración</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Contabilidad</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Finanzas</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Logística</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Producción</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Publicidad y Marketing</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Recursos Humanos</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Ventas</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> </tbody> </table> <p>Dicha matriz deberá contemplar las distintas áreas que existan en el sistema.</p> <p>El nivel de usuario indicado en la matriz podrá tomar los siguientes valores:</p> <ul style="list-style-type: none"> <li>• Nivel 5. No tendrá acceso a este tipo de información.</li> <li>• Nivel 4. Indica que el usuario sólo podrá acceder a los datos de la unidad de negocio a la que esté asociado.</li> <li>• Nivel 3. Indica que el usuario solo tendrá acceso a los datos de las unidades de negocio de la empresa a la que esté asociado.</li> <li>• Nivel 2. Indica que el usuario podrá acceder a los datos de las empresas del mismo rubro al que está asociado.</li> <li>• Nivel 1. Indica que el usuario tendrá acceso a los datos de todas las empresas del Grupo CS.</li> </ul>				Datos detallados	Datos consolidados	KPIs	Administración	Nivel	Nivel	Nivel	Contabilidad	Nivel	Nivel	Nivel	Finanzas	Nivel	Nivel	Nivel	Logística	Nivel	Nivel	Nivel	Producción	Nivel	Nivel	Nivel	Publicidad y Marketing	Nivel	Nivel	Nivel	Recursos Humanos	Nivel	Nivel	Nivel	Ventas	Nivel	Nivel	Nivel
	Datos detallados	Datos consolidados	KPIs																																				
Administración	Nivel	Nivel	Nivel																																				
Contabilidad	Nivel	Nivel	Nivel																																				
Finanzas	Nivel	Nivel	Nivel																																				
Logística	Nivel	Nivel	Nivel																																				
Producción	Nivel	Nivel	Nivel																																				
Publicidad y Marketing	Nivel	Nivel	Nivel																																				
Recursos Humanos	Nivel	Nivel	Nivel																																				
Ventas	Nivel	Nivel	Nivel																																				
Requisitos Relacionados:																																							

*Tabla 151. Requisito R05 instanciado*

Para terminar, vamos a completar y revisar los requisitos para asegurarnos de la calidad del trabajo. Como parte de la revisión realizamos las siguientes acciones:

- Completar los atributos que estaban en blanco.
- Combinar los requisitos R02 y R03 en uno solo.
- Editar la redacción de algunas oraciones.

El resultado obtenido se presenta de la Tabla 152 a la Tabla 156.

# SOLUCIÓN PROPUESTA

Identificador:	RA001	Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:	Alta	Necesidad:	Esencial
Descripción:	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se informará: el nombre de la Unidad de Negocios y/o Empresa a la que pertenece.		
Requisitos Relacionados:	RA002		

Tabla 152. Requisito RA001

Identificador:	RA002	Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:	Alta	Necesidad:	Esencial
Descripción:	<p>Los privilegios dados a un usuario, vendrán informados en el perfil que se le asigne. Los perfiles de acceso al sistema serán descritos como la combinación de dos criterios:</p> <ul style="list-style-type: none"> <li>• Tipos de Usuario</li> <li>• Niveles de Usuario</li> </ul>		
Requisitos Relacionados:	RA 003 RA004		

Tabla 153. Requisito RA002

Identificador:	RA003	Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:	Alto	Necesidad:	Esencial
Descripción:	<p>Para limitar el acceso a los módulos que se definan en la aplicación existirán 5 tipos de usuario:</p> <ul style="list-style-type: none"> <li>• Lectura. Solo podrá acceder a la aplicación en modo lectura, es decir, solo podrá visualizar informes predefinidos ya ejecutados. Este será el perfil del usuario general.</li> <li>• Escritura. Además de poder visualizar los informes predefinidos, podrá acceder al software para ingresar o cargar información en el sistema. Este perfil se utilizará para almacenar nueva información en la base de datos.</li> <li>• Ejecución. Además de poder acceder a la aplicación para visualizar los informes previamente ejecutados, podrá ejecutar y personalizar cada uno de los reportes previamente diseñados por otros usuarios.</li> <li>• Diseño. Además de poder realizar las acciones de los usuarios de tipo Lectura y Ejecución, podrá acceder a la aplicación para crear nuevos reportes para la aplicación.</li> <li>• Administración. Además de poseer los permisos del usuario de Escritura y el de Diseño, tendrá acceso a la parte de administración (seguridad, parametrización...) del Sistema. En este perfil estará el personal especializado de TI.</li> </ul>		
Requisitos Relacionados:	RA002		

Tabla 154. Requisito RA003

## SOLUCIÓN PROPUESTA

Identificador:	RA004	Versión:	01																																				
Tipo:	Acceso al sistema	Autor:	E. Cáceres																																				
Prioridad:	Alta	Necesidad:	Esencial																																				
Descripción:	<p>Los permisos de acceso a la información del sistema serán otorgados mediante la asignación de privilegios en base a la siguiente matriz:</p> <table border="1"> <thead> <tr> <th></th><th>Datos detallados</th><th>Datos consolidados</th><th>KPIs</th></tr> </thead> <tbody> <tr> <td>Administración</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Contabilidad</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Finanzas</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Logística</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Producción</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Publicidad y Marketing</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Recursos Humanos</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> <tr> <td>Ventas</td><td>Nivel</td><td>Nivel</td><td>Nivel</td></tr> </tbody> </table> <p>Dicha matriz deberá contemplar las distintas áreas que existan en el sistema.</p> <p>El nivel de usuario indicado en la matriz podrá tomar los siguientes valores:</p> <ul style="list-style-type: none"> <li>Nivel 5. No tendrá acceso a este tipo de información.</li> <li>Nivel 4. Indica que el usuario sólo podrá acceder a los datos de la unidad de negocio a la que esté asociado.</li> <li>Nivel 3. Indica que el usuario solo tendrá acceso a los datos de las unidades de negocio de la empresa a la que esté asociado.</li> <li>Nivel 2. Indica que el usuario podrá acceder a los datos de las empresas del mismo rubro al que está asociado.</li> <li>Nivel 1. Indica que el usuario tendrá acceso a los datos de todas las empresas del Grupo CS.</li> </ul>				Datos detallados	Datos consolidados	KPIs	Administración	Nivel	Nivel	Nivel	Contabilidad	Nivel	Nivel	Nivel	Finanzas	Nivel	Nivel	Nivel	Logística	Nivel	Nivel	Nivel	Producción	Nivel	Nivel	Nivel	Publicidad y Marketing	Nivel	Nivel	Nivel	Recursos Humanos	Nivel	Nivel	Nivel	Ventas	Nivel	Nivel	Nivel
	Datos detallados	Datos consolidados	KPIs																																				
Administración	Nivel	Nivel	Nivel																																				
Contabilidad	Nivel	Nivel	Nivel																																				
Finanzas	Nivel	Nivel	Nivel																																				
Logística	Nivel	Nivel	Nivel																																				
Producción	Nivel	Nivel	Nivel																																				
Publicidad y Marketing	Nivel	Nivel	Nivel																																				
Recursos Humanos	Nivel	Nivel	Nivel																																				
Ventas	Nivel	Nivel	Nivel																																				
Requisitos Relacionados:	RA002																																						

*Tabla 155. Requisito RA004*

Identificador:	RA005	Versión:	01
Tipo:	Acceso al sistema	Autor:	E. Cáceres
Prioridad:	Media	Necesidad:	Esencial
Descripción:	Se asignarán tipos de usuario con sus respectivos niveles de seguridad a todas las organizaciones implicadas en el sistema.		
Requisitos Relacionados:	RA001		

*Tabla 156. Requisito RA005*

Finalmente, debemos revisar los *criterios de validación* para asegurarnos que ya tenemos la especificación del sistema, es decir, la salida de este proceso.

## SOLUCIÓN PROPUESTA

---

- ¿Los requisitos obtenidos representan la realidad del proyecto? Sí, estos requisitos han sido instanciados teniendo en cuenta las características propias del proyecto.
- ¿Los stakeholders han aprobado los requisitos elicitados? Este criterio está relacionado con la validación por parte del cliente. Como este es un ejemplo, asumimos que los involucrados han aprobado nuestros requisitos.

Por lo tanto, la salida este proceso, serían los requisitos: RA001 (Tabla 152), RA002 (Tabla 153), RA003 (Tabla 154), RA004 (Tabla 155) y RA005 (Tabla 156).

Con este paso, damos por concluida la aplicación de la metodología en este ejemplo, deseando que el lector haya comprendido cada uno de los pasos del proceso de *Indexación y Recuperación* de un patrón.

# **EVALUACIÓN DE LA SOLUCIÓN**

## 4 Evaluación de la solución

En todo proyecto de investigación existe la necesidad ineludible de probar su validez. Para demostrar esta validez de la hipótesis expuesta en el presente trabajo de investigación, planteamos un experimento.

### 4.1 Experimentación

Para realizar la evaluación realizamos un experimento que permitió comprobar la hipótesis planteada al inicio de la presente investigación. Este experimento está descrito a continuación. Tenemos que resaltar que esta sección fue plasmada por el autor de la tesis de manera similar en el artículo: “*Validación de la aplicación de un modelo para la mejora de la ingeniería de requisitos*” (Salazar, et al., 2015) con la colaboración de su asesora.

#### 4.1.1 Metodología de Validación

##### 4.1.1.1 Definición de objetivos y estrategia general

El objetivo del experimento es demostrar que usando una herramienta automática con patrones de requisitos (HCP) basados en MORORE, respecto a otras dos herramientas convencionales (MS-Word (MSW) y Herramientas CASE sin patrones (HCS)), se puede reducir el tiempo de desarrollo y mejorar la calidad de la especificación de los requisitos, en un contexto del desarrollo de proyectos de ingeniería de software con estudiantes de pregrado de diferentes titulaciones afines de una determinada universidad.

El contexto del experimento lo detallamos a continuación:

- El experimento se desarrolló con alumnos de pregrado de tres sedes o titulaciones distintas de la Universidad de Piura (Perú): Análisis de Sistemas de la Escuela Tecnológica Superior (ETS), Ingeniería Industrial y de Sistemas Campus Lima (IIS-Lima) e Ingeniería Industrial y de Sistemas Campus Piura (IIS-Piura). Específicamente, en cada una de las tres sedes, se dicta una asignatura donde se trata la Ingeniería de Requisitos.
- Respecto al historial de los datos para la experimentación, se tomaron desde el año 2009 hasta el 2014 (6 años).
- Para el experimento se escogieron 3 asignaturas similares (una de cada sede) que incluían en su contenido el tema de Ingeniería de Requisitos. En estas asignaturas se encargó a los alumnos relacionados al experimento, desarrollar un trabajo de especificación de requisitos. El trabajo dejado a los alumnos, tuvo las mismas características para todas las sedes:
  - El contenido del trabajo fue el mismo para todas las sedes, incluyó la especificación de requisitos (ver Anexo 4: Contenido del trabajo). Asimismo, todos los alumnos recibieron los mismos temas respecto a la Ingeniería de Requisitos previos a la experimentación en sus respectivas asignaturas.
  - Para cada trabajo se formaron equipos de máximo 5 alumnos.

## EVALUACIÓN DE LA SOLUCIÓN

- Todos los alumnos contaron con los mismos recursos académicos para el desarrollo del trabajo, es decir, todos sin excepción contaron con computadoras, libros especializados, y orientación y guía a cargo del profesor tutor.
- El trabajo se desarrolló siguiendo los lineamientos (procesos, formatos, etc.) vistos en clases con asesorías del tutor. Cada equipo de trabajo recibió dos asesorías obligatorias (típicamente eran más) con un tutor (profesional encargado del curso, distinto para cada sede (1 por sede)) antes de la culminación del trabajo. Esto aseguraba un mínimo de estandarización en la aplicación de los conocimientos y técnicas dadas en clase para el desarrollo del proyecto.
- Los profesores usaron los mismos criterios de evaluación para los trabajos en las 3 sedes.
- El trabajo se desarrolló sobre una empresa real del entorno. Cada equipo de trabajo se encargó de buscar una empresa o entidad comercial real, con la finalidad de realizar en ella un proceso de desarrollo de software que hiciera eficiente alguno de sus procesos de información.
- Al inicio, el alcance del trabajo (proyecto), se aprobó por el profesor de la asignatura, para que tuvieran un esfuerzo similar en todos los salones y en las tres sedes. El tutor evaluó los proyectos en base a lo siguiente: Ser de un área específica de una empresa, no de varias áreas; Que abarcará máximo tres procesos de sistemas de información; Que el sistema sea manejado máximo por 5 tipos de usuarios; Que el sistema de información tenga una complejidad homogénea para todos los equipos.
- Para el experimento se utilizaron 3 herramientas de soporte para el desarrollo de la especificación de requisitos (Tabla 157). Específicamente para el experimento, la herramienta CASE con patrones de MORORE fue: swReuser (swReuser, 2017). En esta herramienta se implementó un módulo para reusar patrones de requisitos basados en MORORE. Y luego se crearon un grupo de patrones de requisitos reusables.

Nº	Herramienta
1	Microsoft Word (MSW)
2	Herramienta CASE sin patrones (HCS)
3	Herramienta CASE con Patrones (HCP)

*Tabla 157 Herramientas usadas en el experimento*

- En el experimento participaron 375 alumnos, divididos en 91 equipos de trabajo, divididos en 12 salones y divididos en las 3 sedes. Para definir el tamaño de la muestra sea significativo se utilizó la Fórmula 3 (Angulo, 2007) y el cálculo lo apreciamos en la Fórmula 4:

$$n = \frac{z^2 pq}{e_{max}^2}$$

*Fórmula 3 Fórmula para obtener el tamaño de la muestra*

## EVALUACIÓN DE LA SOLUCIÓN

Donde:

- $z$ : Es la confiabilidad que se desea tener, en este caso se desea una confiabilidad del 95%, y con la ayuda de la tabla que proporciona áreas bajo la curva normal, se puede determinar el valor de  $z$ , 1.96.
- $pq$ : será el 0.25 en el peor de los casos, este es máximo valor que puede tomar, ya que no se conoce el valor de la proporción.
- $e_{\text{max}}$ : es el máximo error muestral que se estuvo dispuesto a cometer, el 5%.

$$n = \frac{1.96^2 \cdot 0.25}{0.05^2} = 384.16$$

*Fórmula 4 Cálculo del tamaño de la muestra.*

Por tanto, fueron necesarios 385 alumnos, sin embargo, esta investigación tuvo a 375 alumnos, debido a que los 10 alumnos restantes requerirían un semestre adicional de prueba, lo que extendería la investigación, por tanto, se decidió trabajar con  $n=375$  y un  $e_{\text{max}}=5.061\%$  manteniendo una población que permita obtener resultados fiables.

- A cada salón (a sus respectivos alumnos) se asignó aleatoriamente una herramienta, excepto cuidando que en cada sede se utilizaran las tres herramientas. Cada herramienta la usaron 4 salones distintos (Tabla 158). Asimismo, hay que resaltar que, cada herramienta tiene una cantidad equiparable de equipos, pero sobre todo de alumnos (125, 124 y 126 alumnos respectivamente por herramienta).

GRUPO	SEDE	SALÓN	EQUIPOS	ALUMNOS
1 (MSW)	ETS	ETS - 2013 II	5	21
	IIS-PIURA	ADS – 2012 I	14	54
	IIS-PIURA	ADS – 2009 I	4	14
	IIS-LIMA	ADS – 2013 I (Lima)	8	36
		TOTAL	31	125
2 (HCS)	IIS-LIMA	ADS – 2014 I (Lima)	6	22
	ETS	ETS – 2012 I	5	18
	IIS-PIURA	ADS – 2011 I	13	60
	IIS-PIURA	ADS – 2010 I	6	24
		TOTAL	30	124
3 (HCP)	ETS	ETS – 2013 I	3	12
	ETS	ETS – 2014 I	8	32
	IIS-PIURA	ADS – 2013 I	9	41
	IIS-LIMA	ADS – 2009 II (Lima)	10	41
		TOTAL	30	126

*Tabla 158 Distribución de los alumnos por herramienta.*

Se podría objetar que éticamente es incorrecto que la nota de los alumnos dependa de una variable que no dependa de su propio desempeño y méritos. Pero como mencionamos, en una misma aula se utilizaba la misma herramienta que hasta antes de la investigación no habíamos comprobado cual era mejor. Además, rotamos aleatoriamente el uso de las tres herramientas en las tres sedes, aunque en salones de diferentes



cuatrimestres (Tabla 158). Asimismo, entre los profesores tutores se conversó sobre una compensación en el caso de uso de las herramientas que al parecer fueran desfavorables. Esta compensación quedó a criterio de cada profesor. Otro punto que señalar es que la “nota de requisitos” representaba el 40% del trabajo de requisitos. Y el trabajo de requisitos representaba 1.5 de peso ponderado de un total de 15 para aprobar la asignatura.

Tenemos que resaltar que uno de los tutores de una de las sedes (IIS-Piura) es el autor de la presente investigación, los otros dos tutores de las otras sedes no pertenecían al equipo de trabajo de la investigación. Y para minimizar el posible sesgo de valoración de la calidad de los trabajos respecto a la investigación, rotamos aleatoriamente cada cuatrimestre y cada año la asignación de la herramienta a usar, como ya lo mencionamos.

### 4.1.2 Validación de Hipótesis

En este apartado veremos por separado la demostración de las dos hipótesis planteadas en la introducción de la tesis doctoral: la hipótesis referente a las diferencias de *tiempo* en el desarrollo y a las diferencias de *calidad* de la especificación de requisitos en un proyecto respecto al uso de una determinada herramienta.

#### 4.1.2.1 Hipótesis de Calidad

##### 4.1.2.1.1 Estructura del experimento de Calidad

Para evaluar el parámetro de calidad tomamos como datos las notas<sup>13</sup> de los trabajos de ingeniería de requisitos de los 375 alumnos. Específicamente las notas referente a la calificación de la especificación de requisitos (Anexo 1: grupos de alumnos con calificaciones).

A continuación, especificamos los lineamientos del caso de estudio a realizar por todos los alumnos en la asignatura:

El trabajo del cuatrimestre constó de dos partes, la primera fue el Análisis del Sistema, en donde se desarrolló la Ingeniería de Requisitos, y la segunda fue el Diseño de dicho Sistema, esta última no fue necesaria para el presente estudio.

Este trabajo parte de la idea de solucionar un problema u optimizar un proceso en una determinada organización. El equipo de trabajo se contactó con la empresa y solicitaron algunas reuniones entre clientes o interesados (gente de la empresa) y analistas (equipo de trabajo). En estas reuniones los analistas deberían entender el problema que tenían los clientes en su organización, y plantear los requisitos que llevaron a cabo la solución del problema u optimización del proceso.

Con la asesoría de su respectivo tutor, cada equipo de trabajo creó y especificó un conjunto de requisitos para el análisis del nuevo sistema mediante el uso de una herramienta específica (Microsoft Word (MSW), una Herramienta Case Sin patrones (HCS) o una Herramienta Case con Patrones (HCP)).

---

<sup>13</sup> En Perú se usa la escala de calificación vigesimal: las notas van en un rango de 0 a 20.

## EVALUACIÓN DE LA SOLUCIÓN

Cada equipo de trabajo fue evaluado por el profesor a cargo, el cual asignó una nota a cada indicador (Contenido, Requisitos, Exposición y Respuestas a las preguntas) según los criterios de la Tabla 159 y Tabla 160. La nota general del trabajo se obtenía del promedio ponderado respectivo.

Presentación escrita del proyecto (60%)	Contenido (20%)	-Contenido completo. -Apariencia (uniformidad de formato, justificación de párrafos, etc.). -Redacción estructurada, comprensible, etc. (una idea por párrafo, frases cortas, etc.)
	Requisitos (40%)	-Factores e indicadores de Calidad (cualidades de los requisitos). -Forma de especificación.

Tabla 159 Criterios de Evaluación para la presentación escrita.

Presentación oral del proyecto (40%)	Exposición individual (20%)	-Elocuencia. -Dominio del tema: mensaje claro, enfocado y sintético. -Orden.
	Respuesta a las preguntas (20%)	-Respuestas razonadas. -Conocimiento del trabajo.

Tabla 160 Criterios de evaluación para la presentación oral

La distribución de la calificación, como se observa en la Tabla 159 y Tabla 160, se consolida en la Tabla 161.

Grupal	60%	A + B + C	Nota final
Individual	40%	D + E	

Tabla 161 Suma de los porcentajes de los criterios de evaluación

Gracias a los indicadores expuestos, escogimos para el experimento solo la nota del desarrollo de los requisitos, que representa la calidad del trabajo realizado por cada alumno; si la nota fue muy alta, la calidad fue muy buena, de lo contrario si la nota fue muy baja, el trabajo de requisitos tuvo una mala calidad. Esta calificación fue obtenida en base a los factores y métricas presentes en MORORE, la cual fue estudiada por cada uno de los tutores.

En total se evaluaron a cuatro salones de “ETS”, a cinco de “IIS-Piura” y tres de “IIS-Lima” de esta manera evaluamos un total de 12 aulas. Pero fueron 4 salones por cada herramienta ((Tabla 158), (Anexo 1: grupos de alumnos)).

### 4.1.2.1.2 Experimento y Resultados de Calidad

Para realizar el análisis estadístico de los datos que presentamos a continuación, se ha utilizado el software estadístico Statgraphics (Statgraphics, 2017). Este software, a través del análisis estadístico de las notas, permitió demostrar la validez de la hipótesis.

Primero se ha realizado el análisis de la hipótesis planteada respecto a la calidad. Este análisis consiste en contrastar estadísticamente, mediante un análisis de la varianza (Peña, 2010), si el tipo de herramienta utilizada influye en la nota media obtenida por cada grupo de alumnos. En este contexto, los alumnos de cada grupo son considerados como una muestra de cierta población. A través del análisis de la varianza se desea saber

## EVALUACIÓN DE LA SOLUCIÓN

si las poblaciones son diferentes, a partir de la información de sus muestras. Concretamente, se desea analizar si las diferencias encontradas en las medias muestrales de cada grupo son estadísticamente significativas, y por tanto generalizables a la población. En este caso, la conclusión es que la herramienta utilizada tiene efecto en la nota obtenida. De no resultar significativas esas diferencias, se concluiría que éstas son puramente aleatorias, debidas al azar del muestreo, siendo entonces las medias poblacionales iguales para cada tipo de herramienta.

El contraste que se plantea en el análisis de la varianza es el siguiente. Como hipótesis nula ( $H_0$ ) y alternativa ( $H_1$ ) se tiene:

$H_0$ : la nota media poblacional es la misma en cada grupo

$H_1$ : la nota media de algún grupo es diferente.

El análisis se realiza a partir de las notas de Requisitos de cada alumno. El resultado del contraste se resume en el p-valor (Peña, 2010). El p-valor es un valor numérico entre 0 y 1. Si el p-valor es mayor que el nivel de significación  $\alpha$  que se desea utilizar, habitualmente  $\alpha = 0.05$ , se acepta  $H_0$ . En caso contrario, se rechaza  $H_0$ , aceptándose entonces  $H_1$ . El p-valor puede interpretarse así como un indicador del respaldo que dan los datos a  $H_0$ . Más concretamente, el p-valor mide la probabilidad de encontrar una diferencia de medias como la vista en los datos cuando  $H_0$  sea cierta.

Con la muestra obtenida el p-valor es muy bajo (ver Anexo 5: Análisis Estadístico). Por consiguiente, rechazamos  $H_0$ , y concluimos que el tipo de herramienta influye en la calificación obtenida. El análisis de la varianza proporciona un coeficiente de determinación de valor  $R^2 = 24.8\%$  que se interpreta como que el tipo de herramienta explica el 24.8% de la variabilidad de las notas por Requisito.

Para analizar el detalle de las diferencias de medias encontradas, construimos intervalos de confianza para la media poblacional de NotaReq por cada tipo de herramienta utilizada. Los intervalos se realizaron mediante el método de Bonferroni

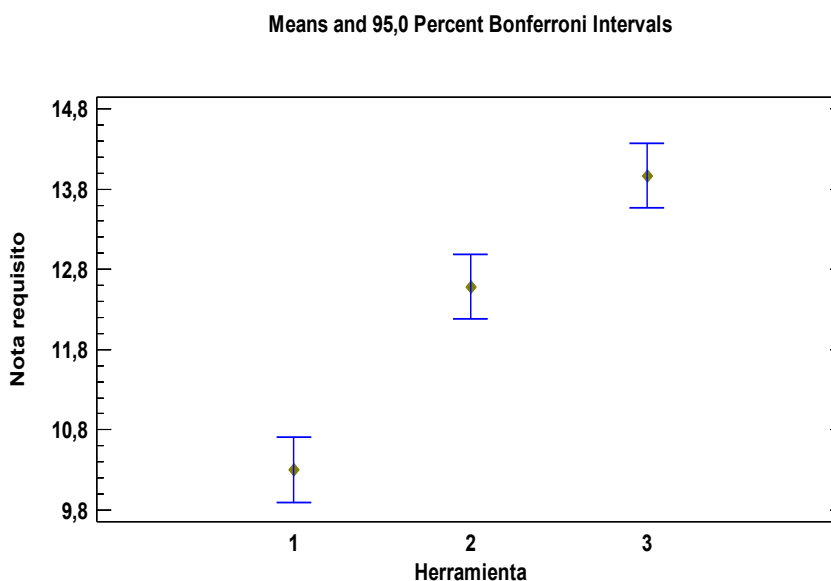


Figura 28 Intervalos de Confianza al 95% - Método de Bonferroni

## EVALUACIÓN DE LA SOLUCIÓN

---

(implementado en Statgraphics), y con un nivel de confianza conjunto del 95%. Este tipo de intervalos se denomina ‘simultáneos’. En estos intervalos, los centros son las medias muestrales para cada grupo, y los intervalos alrededor de ellos están contruidos de tal manera que el nivel de confianza especificado no es para cada uno de ellos sino para la región formada por el conjunto de ellos. De esta forma, lo solapamientos pueden interpretarse como que puede aceptarse la hipótesis nula de que ambas medias poblacionales son iguales.

Si, por ejemplo, los intervalos de las medias  $\mu_1$  (Herramienta 1) y  $\mu_2$  (Herramienta 2) se solapan, aceptaremos la hipótesis nula de que ambas medias son iguales, es decir, aceptamos:  $H_0: \mu_1 = \mu_2$ , con un nivel de significación del 5%. Si, por el contrario, los intervalos no se solapan, rechazamos  $H_0$  y aceptamos  $H_1: \mu_1 \neq \mu_2$ . Los intervalos de Bonferroni nos permiten comparar las diferentes medias asegurando un nivel de confianza del 95%. En ese caso, se aprecia que las tres medias son significativamente diferentes y la mejor herramienta es la 3. Los alumnos que utilizaron la herramienta 3 obtuvieron por término medio 1.4 puntos más que los que utilizaron la herramienta 2, y éstos 2.3 más que los que utilizaron la herramienta 1. Por tanto, usar la herramienta 3 supuso por término medio 3.7 puntos más que usar la herramienta 1.

Los diferentes alumnos no sólo se han diferenciado en el tipo de herramienta utilizada, sino que se diferencian en otros factores que podrían haber influido en la nota de Requisito. Para ver la posible influencia que otras variables explicativas puedan tener, se procedió a realizar un análisis de regresión múltiple (ver 7.5 Anexo 5: Análisis Estadístico). Estas variables con posible influencia son:

- Sede: referidas a las 3 titulaciones relacionadas al experimento de la misma Universidad que funcionan en lugares distintos.
- Tamaño de equipo de alumnos formado: cantidad de alumnos que formaron un grupo para realizar el trabajo.
- Índice Académico del alumno: promedio ponderado del alumno de toda su titulación.
- Salón: nombre del grupo de alumnos de un aula completa de un cuatrimestre y de una sede determinada.

En este caso, del análisis estadístico respectivo, se obtiene que, aparte de la herramienta, las únicas variables significativas para explicar diferencias en la nota de Requisito son: el índice académico de los alumnos, así como pertenecer al salón 9 o a la sede 2. Estas tres variables tienen un efecto mucho menor sobre la nota que el tipo de herramienta. Estas tres variables junto con la herramienta explican un 30.2% de la variabilidad de la NotaReq. De este total solo la herramienta explicaba el 24.8%, por lo tanto, la sede 2, el índice académico y el salón 9 explican solo un 5,4%. Evidenciamos que la herramienta es la principal variable explicativa de la NotaReq.

## EVALUACIÓN DE LA SOLUCIÓN

---

### **4.1.2.1.3 Conclusiones respecto a la calidad**

Las conclusiones que se obtiene del análisis de regresión respecto a la calidad, son las siguientes:

- Para una misma sede, mismo salón, y un mismo índice académico, la nota de 'NotaReq' usando la herramienta 1 es 3.84 puntos inferior por término medio que usando la herramienta 3.
- Para una misma sede, mismo salón y un mismo índice académico, la nota de 'NotaReq' usando la herramienta 2 es 1.72 puntos inferior por término medio que usando la herramienta 3.
- Para una misma herramienta, salón, y sede, alumnos con un punto más de índice académico tienen 0.25 puntos más en NotaReq por término medio.
- Para una misma herramienta e índice académico, en la sede 2 obtuvieron por término medio 0.83 puntos más en NotaReq.

### **4.1.2.2 Hipótesis de Tiempo**

#### **4.1.2.2.1 Estructura del experimento de Tiempo**

Para evaluar el parámetro de tiempo se tomó la información proveniente de las respuestas a una encuesta efectuada a los 375 alumnos que realizaron el trabajo de ingeniería de requisitos (Anexo 3: detalle de las respuestas de cada alumno).

Para este experimento se usó también la misma estructura que para el parámetro calidad, es decir, los alumnos fueron los mismos descritos en la Tabla 158.

A continuación, especificamos los lineamientos tomados en cuenta para tomar los datos para el experimento referente al tiempo.

Se diseñó una encuesta cuyo objetivo principal fue obtener información del alumno referente al tiempo que tardó en desarrollar el trabajo de ingeniería de requisitos. El formato de la encuesta está en el Anexo 2: encuesta de métricas de requisitos (personal). En forma específica, en la encuesta principalmente se abarcaron los siguientes puntos:

- La herramienta que utilizó el alumno para el desarrollo del trabajo.
- Según las especificaciones de tiempo.
  - Tiempo promedio para crear un requisito y una lista de requisitos con la herramienta respectiva.
  - Tiempo promedio para modificar un requisito o varios requisitos.
- Según la gestión de los requisitos.
  - Número de requisitos finales que tuvo en total el trabajo semestral.
  - Si hizo la modificación de algún requisito, cuánto tiempo tardó en modificar la lista de requisitos y cuantos requisitos fueron cambiados.

## EVALUACIÓN DE LA SOLUCIÓN

---

- Número de requisitos eliminados.

Al inicio del trabajo de ingeniería de requisitos, se especificó a los alumnos los lineamientos del caso de estudio del trabajo y de la encuesta a desarrollar.

A cada alumno, al finalizar el trabajo de ingeniería de requisitos, se les pidió que llenaran la encuesta respectiva en forma anónima. Esta encuesta fue realizada vía WEB, en su mayoría, utilizando el portal “encuestafácil.com” (Encuestafacil, 2017).

### **4.1.2.2.2 Experimento y Resultados de Tiempo**

Las encuestas fueron realizadas con éxito en el periodo de tiempo establecido. Se obtuvo la información referida a tiempo y cantidad de creación, modificación y eliminación de requisitos del trabajo.

De igual manera que para el análisis estadístico respecto a la calidad, se ha utilizado el software estadístico Statgraphics (Statgraphics, 2017). Este software, a través del análisis estadístico de los datos obtenidos de las encuestas, permitió demostrar la validez de la hipótesis referente al tiempo. Con este fin, se ha realizado un análisis de regresión múltiple con variables binarias (ver 7.5 Anexo 5: Análisis Estadístico). El análisis estadístico permitió saber si la influencia de una herramienta es significativa respecto a las medias de los tiempos que se tardaron los alumnos al trabajar con los requisitos (crear, modificar y eliminar).

A continuación, presentamos los resultados del análisis estadístico de los datos obtenidos en las encuestas realizadas a los alumnos, por cada pregunta relevante respecto al tiempo.

#### **4.1.2.2.2.1 Resultados con respecto al tiempo de creación de un requisito**

Como ya comentamos, para ver la posible influencia que la variable herramienta pueda tener en el tiempo promedio que tardó un alumno en crear un requisito, se procedió a realizar un análisis de regresión múltiple (ver 7.5 Anexo 5: Análisis Estadístico).

Del análisis estadístico respectivo, se obtiene que hay diferencias significativas en el tiempo medio de creación de un requisito para los tres tipos de herramientas, y que para la herramienta 3 el tiempo es el menor.

El tiempo medio para la realización de un requisito con la herramienta 3 es 9.98 minutos, con la herramienta 2 se demora más tiempo, 18.3 minutos y con la herramienta 1 se demoran aún más tiempo, 43.8 minutos. Asimismo, se obtiene que la herramienta utilizada explica el 58% de las diferencias de tiempo.

#### **4.1.2.2.2.2 Resultados con respecto al tiempo de creación de toda la lista de requisitos**

Después de realizar el análisis estadístico respectivo (ver 7.5 Anexo 5: Análisis Estadístico), se obtiene que también hay diferencias significativas entre la herramienta 3 y las otras dos.

Para los alumnos, el tiempo medio de realización de toda la lista de requisitos con la herramienta 3 es 399 minutos, con la herramienta 2, en media, se demora 168.4 minutos más que con la herramienta 3 y con la herramienta 1 en media se demora 248.9 minutos

## EVALUACIÓN DE LA SOLUCIÓN

---

más que con la herramienta 3. Asimismo, se obtiene que la herramienta explica el 16.97% de las diferencias de tiempo.

### **4.1.2.2.2.3 Resultados con respecto a la cantidad de requisitos creados**

Después de realizar el análisis estadístico respectivo (ver 7.5 Anexo 5: Análisis Estadístico), nuevamente se obtiene que hay diferencias significativas entre las tres herramientas. Siendo con la herramienta 3 con la que se crean más requisitos.

Para los alumnos, en media, con la herramienta 3 se crean 71.9 requisitos. Con la herramienta 2, en media, se crean 16.7 requisitos menos que con la herramienta 3. Y con la herramienta 1, en media se crean 23.2 requisitos menos que con la herramienta 3. Asimismo, la herramienta explica el 41.1% de las diferencias en el número de requisitos creados.

### **4.1.2.2.2.4 Resultados con respecto al tiempo de modificación de la lista de requisitos**

Después de realizar el análisis estadístico respectivo (ver 7.5 Anexo 5: Análisis Estadístico), se obtiene que también hay diferencias significativas entre la herramienta 3 y las otras dos.

Para los alumnos, el tiempo medio de demora en modificar los requisitos con la herramienta 3 es de 170 minutos. Con la herramienta 2 se demora más tiempo, un total de 211 minutos, y con la herramienta 1 se demoran aún más tiempo, un total de 239 minutos. Asimismo, se puede calcular que la herramienta utilizada explica el 4.8% de las diferencias de tiempo.

### **4.1.2.2.2.5 Resultados con respecto a la cantidad de requisitos que fueron modificados**

Después de realizar el análisis estadístico respectivo (ver 7.5 Anexo 5: Análisis Estadístico), en este caso, se obtiene que las herramientas 2 y 3 son similares, es decir, que no hay diferencias significativas. Hay diferencias significativas, no obstante, entre estas dos herramientas y la herramienta 1.

Para los alumnos, en media, la cantidad de requisitos modificados con las herramientas 2 o 3 es de 17.47 requisitos, mientras que con la herramienta 1, en media, se tienen que modificar 3.7 requisitos más que con cualquiera de las otras dos herramientas.

### **4.1.2.2.2.6 Resultados con respecto al tiempo de cambio de un requisito**

Para realizar un análisis mejor detallado, para este acápite se ha construido una variable que es el “*tiempo invertido por cambio de un requisito*” (*(tiempo de modificación de toda la lista de requisitos) / (número de requisitos modificados)*) (ver 7.5 Anexo 5: Análisis Estadístico).

Para los alumnos, el tiempo medio de demora en modificar un requisito con la herramienta 3 es de 9.97 minutos. Con la herramienta 1 se demora más tiempo, 12.3 minutos, y con la herramienta 2 se demoran algo más, 13.3 minutos. De este mismo análisis estadístico, obtuvimos que la diferencia entre los tiempos de cambio de un

## EVALUACIÓN DE LA SOLUCIÓN

requisito utilizando de la herramienta 1 o 2 no resulta significativa. Ambas herramientas son iguales en este aspecto (ver 7.5 Anexo 5: Análisis Estadístico).

### 4.1.2.2.7 Resultados con respecto a la cantidad de requisitos eliminados

Después de realizar el análisis estadístico respectivo (ver 7.5 Anexo 5: Análisis Estadístico), se obtiene que no hay diferencias significativas entre las herramientas 2 y 3, pero sí las hay entre estas dos herramientas y la herramienta 1.

En media, con la herramienta 3 y 2 se eliminaron 7.4 requisitos. Con la herramienta 1, en media, se eliminaron 3.2 requisitos más que con la herramienta 3. Asimismo, se concluye que la herramienta explica el 7.9% de las diferencias de requisitos eliminados.

### 4.1.2.2.3 Conclusiones respecto al tiempo

Como ayuda a la exposición de las conclusiones, en la Tabla 162, exponemos el resumen de los resultados del análisis estadístico de las variables relevantes respecto a la hipótesis de tiempo.

Variables	Herramienta 1	Herramienta 2	Herramienta 3	% de influencia
Tiempo de creación de un requisito	43.8 min.	18.3 min.	9.8 min.	58%
Tiempo de creación de todos los requisitos	647.9 min.	567.4 min.	399 min.	16.97%
Cantidad de requisitos creados	48.7 req.	55.2 req.	71.9 req.	41.10%
Tiempo de modificación de la lista de requisitos	239 min.	211 min.	170 min.	4.80%
Cantidad de requisitos modificados	21.17 req.	17.47 req.	17.47 req.	7.20%
Tiempo de modificación de un requisito	12.3 min.	13.3 min.	9.97 min.	2.80%
Cantidad de requisitos eliminados	10.6 req.	7.4 req.	7.4 req.	7.90%

*Tabla 162 Resumen del análisis estadístico de las variables respecto al tiempo*

En consecuencia, las conclusiones que se obtiene del análisis de regresión respecto al tiempo, son las siguientes:

- El tiempo medio para crear un requisito, con la herramienta 3 (9.8 min.) es menor que con la herramienta 1 (34 minutos más) y la herramienta 2 (8.5 minutos más).
- El tiempo medio para crear la lista de requisitos, con la herramienta 3 (399 min.) es menor que con la herramienta 1 (248.9 minutos más) y la herramienta 2 (168.4 minutos más).
- Aparte de que el alumno se demora menos en crear requisitos con la herramienta 3, también crea más requisitos (71.9 req.) que con la herramienta 1 (23.2 req. menos) y la herramienta 2 (16.7 req. menos).
- El tiempo medio para modificar requisitos, con la herramienta 3 (170 min.) es menor que con la herramienta 1 (69 min. más) y la herramienta 2 (41 min. más).



## EVALUACIÓN DE LA SOLUCIÓN

---

- La cantidad de requisitos modificados, con la herramienta 3 (17.47 req.) es menor que con la herramienta 1 (3.7 req. más). Pero, en este caso, con la herramienta 2 no hay diferencia en la cantidad de requisitos modificados.
- El tiempo medio para modificar un requisito, con la herramienta 3 (9.97 min.) es menor que con la herramienta 1 (2.33 min. más) y la herramienta 2 (3.33 min. más).
- La cantidad de requisitos eliminados, con la herramienta 3 (7.4 req.) es menor que con la herramienta 1 (3.2 req. más). Pero, en este caso, con la herramienta 2 no hay diferencia en la cantidad de requisitos eliminados.

# CONCLUSIONES

### 5 Conclusiones, aportaciones y trabajos futuros

En el presente capítulo presentamos las conclusiones que obtuvimos después de realizar la presente investigación. Estas conclusiones están acreditadas en el documento de la tesis doctoral. Asimismo, en esta sección puntualizamos las aportaciones originales del trabajo, así como las limitaciones del ámbito de la investigación realizada. Y para finalizar expondremos posibles trabajos de investigación tomando como referencia la presente tesis doctoral, y también las actividades que permitieron la aceptación de la presente propuesta en la comunidad científica relacionada.

#### 5.1 Discusión y conclusiones

Desde un principio, el presente trabajo de investigación ha tenido la intención de mejorar la Ingeniería de Requisitos y por consiguiente todo el proceso de desarrollo de software. Para ello, propusimos la aplicamos técnicas de reúso en el ámbito de la ingeniería de requisitos para optimizar, sobre todo, el tiempo de desarrollo y la calidad del producto. Esta aplicación quedó plasmada en la propuesta denominada: Modelo de Requisitos Orientado al Reúso Efectivo (MORORE).

Por medio del análisis del **estado del arte** para el presente trabajo de investigación, pudimos evidenciar que diversos autores coinciden en que la ingeniería de requisitos es una de las fases más importantes del proceso de desarrollo de software, y que los principales problemas de la ingeniería de software se originan en una inadecuada especificación y gestión de los requisitos. Asimismo, en la literatura revisada hemos encontrado muchas propuestas que pretenden la mejora de la Ingeniería de Requisitos.

Entre las propuestas de mejora a la Ingeniería de Requisitos tenemos: diferentes formas de especificar, organizar y gestionar requisitos, algunos alcances sobre la calidad de los requisitos, trabajos con la aplicación de técnicas de reúso en el ámbito de la ingeniería de requisitos, entre otras técnicas y metodologías. La revisión del estado del arte sirvió como base para el desarrollo del Modelo de Requisitos Orientado al Reúso Efectivo – MORORE, que representa el logro del objetivo planteado.

MORORE fue desarrollado en su totalidad. Es decir, se creó la representación de los tres niveles de representación de activos de la ingeniería de requisitos, se creó el modelo de calidad con sus respectivos factores y métricas, y por último se definió el proceso para indexar y recuperar activos de requisitos.

La validez de la hipótesis fue comprobada con un experimento. Este fue realizado con alumnos (como representantes de ingenieros noveles), los cuales desarrollaron un trabajo de ingeniería de requisitos. Se utilizaron 3 tipos de herramientas las cuales fueron evaluadas, una de ellas con una implementación de patrones de requisitos de MORORE. Los proyectos fueron realizados en empresas reales. Y la validez de los resultados obtenidos para un propósito general son evidenciados por la cantidad de la población (de 375 alumnos) y la información recogida a lo largo de 6 años. Asimismo, brindamos la información para la reproducibilidad del experimento.

## CONCLUSIONES

---

Las conclusiones específicas referidas al experimento realizado las exponemos a continuación. Para el efecto, dividimos en dos secciones estas conclusiones: las referidas a la hipótesis de mejora de la calidad (datos de las notas de los trabajos) y luego las referidas a la hipótesis de reducción del tiempo (datos de encuestas realizadas).

Respecto al parámetro de calidad, del experimento se obtuvo lo siguiente:

- Se evaluó la influencia de una determinada herramienta sobre las calificaciones obtenidas por los alumnos en el trabajo realizado. Y de los resultados del experimento podemos concluir, en forma general, que al utilizar una herramienta CASE, con una implementación de patrones de requisitos de MORORE (HCP), se obtienen especificaciones de requisitos de mejor calidad que al utilizar MS-Word (MSW) o una herramienta CASE convencional (HCS) en un nuevo proyecto. Esto se evidencia pues, las notas medias obtenidas respecto al empleo de cada herramienta son significativamente diferentes, además, con HCP se obtuvieron mejores notas que con HCS en torno a un 10% superior y mucho mejor que con MSW en torno a un 26% superior.
- Asimismo, se evaluaron otros factores (sede, tamaño de equipo, índice académico y salón) que podían tener influencia sobre las notas. Y se obtuvo que algunos de estos factores tuvieron influencia significativa sobre las notas (índice académico, salón 9 y sede 2). También se obtuvo que, estos últimos factores junto con la herramienta explican un 30.2% de la variabilidad de la nota. Pero solo la herramienta explica un 24.8% y el resto juntos explica solo un 5.4%. Por lo tanto, la variable herramienta influye en mayor medida sobre la nota que otros factores.

Respecto al parámetro del tiempo, del experimento se obtuvo lo siguiente:

- Se evaluó la influencia de una determinada herramienta sobre los tiempos de demora de los alumnos al realizar el trabajo de ingeniería de requisitos. Y de los resultados del experimento podemos concluir, en forma general, que al utilizar una herramienta HCP se obtienen tiempos menores al trabajar con requisitos (al crear y modificar requisitos) que al utilizar MSW o HCS en un nuevo proyecto. Para crear la lista de requisitos con HCP tomó menos tiempo que con HCS en torno a un 42% inferior y también menos tiempo que con MSW en torno a un 62% inferior. Asimismo, para modificar la lista de requisitos con HCP tomó menos tiempo que con HCS en torno a un 24% inferior y también menos tiempo que con MSW en torno a un 40% inferior.
- Asimismo, del experimento se obtiene que al utilizar la HCP se crean mayor cantidad de requisitos que con HCS en torno a un 23% superior y mayor que con MSW en torno a un 32% superior; con HCP se modifican menos requisitos que con MSW entorno a un 21% inferior; y con HCP se eliminan menos requisitos que con MSW entorno a un 43% inferior.

En resumen, a partir del experimento concluimos que el factor herramienta influye significativamente sobre la calidad (efectividad) y el tiempo de desarrollo (eficiencia) de un proyecto de especificación de requisitos. Siendo la herramienta CASE con patrones (HCP) la mejor para desarrollar requisitos de calidad y reducir el tiempo de desarrollo de una especificación de requisitos. En consecuencia, a lo realizado, hacemos real una propuesta para la mejora de la ingeniería de requisitos. Y por la importancia de la ingeniería de requisitos para el desarrollo de sistemas informáticos, sostenemos que

## CONCLUSIONES

---

estamos contribuyendo también al mejoramiento de todo el proceso de desarrollo de software.

### 5.2 Aportaciones originales

La presente tesis doctoral propone la implementación y aplicación de **tres niveles de activos software**: requisitos individuales, patrones de requisitos y estructuras de tipos de requisitos. Estos niveles pueden ser reusados de forma independiente o combinados según más convenga a una situación determinada. Es decir, por su naturaleza los activos software son indexados de forma independiente y de igual forma recuperados. Todo esto a pesar que puede existir ciertas redundancias entre los activos de diferentes niveles almacenados. Por ejemplo, un patrón de requisitos puede contener un mismo requisito que esté almacenado independientemente.

Asimismo, proponemos la definición de un modelo que consta de un conjunto de **factores y métricas para evaluar la calidad de los activos** de requisitos de cada nivel de MORORE. Por la naturaleza misma de estos activos, los factores y métricas para definir la calidad pueden variar para cada tipo de activo. Es decir, que para medir la calidad de un requisito usamos determinados factores con sus respectivas métricas específicas, que no son necesariamente los mismos para medir la calidad de un patrón de estructuras de tipos de requisitos.

También tenemos en el presente trabajo de investigación la definición de una **metodología que permita el reúso** dentro de los lineamientos conceptuales de MORORE. Como ya mencionamos esta metodología consta de dos partes esenciales: **la indexación y la recuperación**. Y cada proceso de esta metodología consta de entradas, salidas, herramientas de ayuda y criterios de validación definidos expresamente para el reúso de los activos de MORORE.

### 5.3 Trabajos futuros

En la herramienta CASE **swREUSER** de la marca comercial **The Reuse Company**, se desarrolló un módulo que permite recuperar patrones de requisitos. Está módulo sirvió para poder realizar el experimento para la respectiva comprobación de las hipótesis planteadas. Pero un trabajo próximo es la elaboración de una herramienta que permita dar soporte al proceso genérico de reúso de MORORE, desde la creación de los activos, el control de calidad, el almacenamiento (se sugiere empleando listas relacionadas), la recuperación (se sugiere usando un modelo vectorial) e integración a un nuevo proyecto de los activos de reúso. Esta herramienta permitiría hacer eficiente el proceso de reúso planteado para MORORE.

MORORE es un modelo creado específicamente para el reúso de activos software en el ámbito de la ingeniería de requisitos. Concretamente los activos software de los tres niveles planteados, su respectivo modelo de calidad y el proceso genérico de reúso. Pero esto no impide que este modelo pueda ser extrapolado con éxito a otras fases del proceso de desarrollo de software. Es más, proponemos a otros investigadores el estudio de la aplicación del modelo de requisitos para el reúso efectivo (MORORE) en otros ámbitos del proceso de desarrollo de software. En este sentido, planteamos la generalización del

## CONCLUSIONES

---

*modelo de requisitos orientado al reuso efectivo (MORORE) al genérico modelo de activos software orientado al reuso efectivo (MASORE).*

MASORE, de manera general, tiene las mismas características del modelo motivo de la presente tesis doctoral (MORORE). Pero, entre las diferencias específicas que se tienen, planteamos las siguientes:

- En vez de requisitos, MASORE generaliza a todo tipo de activos software. Estos activos software genéricos podrían instanciarse en activos software específicos dependiendo del ámbito de la fase del proceso de desarrollo de software donde se apliquen.
- Otra diferencia fundamental, son los distintos atributos de especificación que tendrán los activos software de otras fases del proceso de desarrollo de software. Esta diferencia es comprensible por la distinta naturaleza de los activos software de los diferentes ámbitos.
- Otro punto de divergencia, es que, para los distintos niveles de activos software de otros ámbitos, se tendrán que definir sus propios factores y métricas de calidad.

Concluimos entonces, que las ideas planteadas en MORORE pueden ser extrapoladas a otros ámbitos del proceso de desarrollo de software. Pero esta extrapolación tiene que tomar en cuenta todas las diferencias descritas. Por la magnitud de los cambios y trabajo de creación expuestos, planteamos que la aplicación de MORORE a otros ámbitos del proceso de desarrollo de software puede sustentarse como un completo estudio de investigación independiente de la presente tesis doctoral.

La validez del presente trabajo se demuestra frente a técnicas convencionales de desarrollo de requisitos, no frente a otras formas de desarrollo con técnicas de reutilización. Un próximo trabajo de investigación es probar la eficiencia y efectividad de la propuesta de la presente tesis frente a otros trabajos de reuso en la ingeniería de requisitos.

### 5.4 Aceptación

En este acápite mostramos la aceptación, por otras personas e instituciones relacionadas, que ha tenido la investigación mostrada en la tesis doctoral. Todo esto expuesto por medio de: publicaciones, conferencias, proyectos, aplicaciones informáticas y universidades.

#### 5.4.1 Publicaciones

A continuación, tenemos el listado de los resúmenes de las publicaciones que tiene directamente relación con el trabajo de investigación de esta tesis doctoral:

- Génova, G. Fuentes, J. M. Llorens, J. Hurtado, O. Moreno, V. *A framework to measure and improve the quality of textual requirements. In Journal of Requir. Eng. vol. 18(1), pp 25-41. ISSN: 0947-3602. (2013)* (Génova, et al., 2013). Relacionado con la tesis en el punto 3.3 Modelo de calidad. “La mejora de la calidad del software exige controles de calidad desde el inicio del proceso de desarrollo, es decir, la captura y escritura de requisitos. La automatización de las métricas de calidad puede suponer un ahorro considerable frente a las tediosas evaluaciones realizadas de modo manual.

## CONCLUSIONES

---

Presentamos algunos indicadores para medir la calidad de requisitos textuales, así como una herramienta que calcula las medidas de calidad de forma totalmente automática. Queremos resaltar en que el objetivo final es medir para mejorar. Reducir la gestión de la calidad a la obtención de una valoración numérica tropezaría con la oposición frontal de los propios analistas, que no verían en el proceso de medición la ayuda de un consejero, sino un mecanismo policial de penalización. Para evitar esto, los indicadores de calidad deben ante todo señalar defectos concretos y ofrecer sugerencias para mejorar. El resultado final no sólo será una mejora en la calidad de los requisitos, sino también una mejora en las habilidades de escritura de los ingenieros de requisitos”.

- *Hurtado, O. Llorens, J. Génova, G. Fuentes, J.M. Generación de Patrones de Requisitos en una Herramienta CASE: Aplicación al Common Criteria. In Hernán Astudillo & Carla Taramasco, ed., 'CibSE', pp. 21-32. ISBN: 956-7051-07-0. Valparaíso, Chile. (2005) (Hurtado, et al., 2005).* Relacionado con la tesis en el punto 3.2.2 Nivel de patrón de requisitos. “La gestión de requisitos es una labor complicada. Existen estándares generales de requisitos para todo tipo de sistemas de información. La complejidad de estos estándares dificulta su adaptación a un proyecto concreto. Presentamos como ejemplo: la norma “Common Criteria for Information Technology Security Evaluation” sobre requisitos de seguridad para todo tipo de tecnologías de la información, la cual es cada vez más requerida a nivel mundial. Este carácter universal de la norma, la hace compleja de entender y usar. Esto dificulta el proceso de desarrollo de software. Se puede cambiar esta situación desfavorable si usamos la idea de patrones en los estándares y los aplicamos mediante una herramienta CASE a proyectos específicos. Este trabajo define un modelo de patrón de requisitos que se implementa en una herramienta CASE. Para el presente informe el modelo soporta al estándar “Common Criteria for Information Technology Security Evaluation”. Creemos que esta integración de tecnologías para reusar requisitos mejorará la gestión de requisitos y consecuentemente todo el proceso de desarrollo de software”.
- *Hurtado, O. Fraga, A. Hernanz, I. Llorens, J. Metamodel of requirements' types classifications for improving the software development process (SDP): an ontological approach. IASTED International Conference on Software Engineering and Applications (SEA2006). Dallas (Texas), USA. ISBN Hardcover: 0889866422 / CD: 088986599X. (2006) (Hurtado, et al., 2006).* Relacionado con la tesis en el punto 3.2.3 Nivel de estructura de tipos de requisitos. “En la actualidad existen muchas clasificaciones de tipos de requisitos ofrecidas por los estándares, metodologías o aplicaciones. Típicamente una herramienta CASE no soporta diferentes estructuras de tipos de requisitos. Es importante no confundir estas clasificaciones de tipos de requisitos con clasificaciones de requisitos específicos. La solución propuesta a este problema es un metamodelo para representar cualquier clasificación de tipos de requisitos y las relaciones entre ellos en un primer paso y una ontología capaz de representar el conocimiento relacionado con las clasificaciones de los tipos de requerimientos de estándares como la ESA, IEEE, Common Criteria y el PAO. Finalmente, una aspiración del trabajo implementar este metamodelo en una herramienta CASE como swREUSER. El objetivo de este trabajo fue la realización de un modelo ontológico que facilite la creación y reutilización del conocimiento respecto a los tipos de requisitos. Este metamodelo ayudará a la gestión de requisitos en los proyectos de desarrollo de software específicos y promoverá el reúso de las clasificaciones de tipos de requisitos, y la optimización de una herramienta CASE

## CONCLUSIONES

---

comercial. Por consiguiente, resultado la mejora del proceso de desarrollo de software (SDP) en el campo de la Ingeniería de Requisitos (RE)”.

- *Hurtado, O. Quinde, M. Martín, D. Particularización dinámica de plantillas para la gestión de requisitos. In Journal of Tecnología y Desarrollo. vol. 7(1), pp 121-126. ISSN: 1819-4579. (2009) (Hurtado, et al., 2009).* Relacionado con la tesis en el punto 3.2.1.1 *Plantilla de requisitos configurable*. “Muchos estudios indican que los problemas del software se originan en la ingeniería de requisitos, especialmente en la gestión de requisitos. En tal sentido, las herramientas de gestión de requisitos mejoran notablemente el proceso de esta ingeniería. Las plantillas de gestión de requisitos de estas herramientas, en la mayoría de los casos, no coinciden con las requeridas para un proyecto específico. En consecuencia, estas plantillas dificultan la gestión efectiva de requisitos y, por tanto, afectan a todo el proceso de desarrollo de software. Nuestro estudio propone el desarrollo de un modelo que permita la personalización dinámica de plantillas para la gestión de requisitos. Este modelo, implementado en una herramienta automática, permitirá al ingeniero responsable personalizar las plantillas de gestión de requisitos para un proyecto determinado. En consecuencia, optimizaremos la gestión de requisitos y por consiguiente mejoraremos todo el proceso de desarrollo de software”.
- *Sánchez, K. Hurtado, O. Génova, G. Modelo para medir la calidad de activos de la ingeniería de requisitos. En actas del 1° Congreso Internacional de Ingeniería: Aporte de la ingeniería al desarrollo sostenible del Perú. Universidad Privada Antenor Orrego de Trujillo. (2015) (Sánchez, et al., 2015).* Relacionado con la tesis en el punto 3.3 *Modelo de calidad*. “En este artículo presentamos un modelo para medir y garantizar la calidad de los activos de la ingeniería de requisitos, como son los requisitos individuales, patrones de requisitos y estructuras de tipos de requisitos. El modelo se basa en un conjunto de factores o cualidades que consideramos un activo de calidad debe presentar; cada cualidad es evaluada de acuerdo a un conjunto de indicadores cuantificables mediante un sistema de métricas. Los valores obtenidos tanto en los factores como en los indicadores permiten identificar cuáles son los aspectos prioritarios que deben ser mejorados en el activo. Sabemos que los principales problemas en ingeniería de software tienen su origen precisamente en la ingeniería de requisitos; concretamente en una deficiente especificación y gestión de requisitos. Por ello, tener los requisitos bien definidos permite mejorar el proceso de la Ingeniería de requisitos y por consiguiente de todo el proceso de desarrollo de software”.
- *Salazar, D., Hurtado, O, Génova, G. Validación de la aplicación de un modelo para la mejora de la ingeniería de requisitos. En actas del 1° Congreso Internacional de Ingeniería: Aporte de la ingeniería al desarrollo sostenible del Perú. Universidad Privada Antenor Orrego de Trujillo. (2015) (Salazar, et al., 2015).* Relacionado con la tesis en el punto 4.1 *Experimentación*. “En éste artículo se presenta la validación de MORORE (MOdelo de Requisitos Orientado al Reúso Efectivo). Esta validación consiste en la descripción de un experimento para probar la reducción del tiempo y mejora de la calidad en el desarrollo de requisitos de software, aplicando un modelo para el reúso de requisitos software. Esta experimentación fue realizada con grupos de alumnos que desarrollaron proyectos de Ingeniería de Requisitos para sus trabajos de elaboración de software. Estos alumnos usaron distintas vías para el desarrollo de requisitos como: Microsoft Word, una Herramienta Case y una Herramienta Case con Patrones de MORORE. La toma de tiempos al realizar el experimento en el desarrollo



## CONCLUSIONES

---

de los proyectos, y la calidad de los requisitos reflejada en las notas obtenidas por los alumnos, fueron los datos a comparar entre las diferentes vías para el desarrollo de los requisitos. Los resultados obtenidos comprueban que usando los Patrones de MORORE, hace más eficiente y eficaz la Ingeniería de Requisitos. Por ello, se concluye, que el reúso de requisitos usando los patrones de MORORE se disminuye el tiempo de desarrollo y garantiza un mínimo de calidad en la creación de requisitos software”.

### 5.4.2 Conferencias

A continuación, listamos las conferencias y seminarios dictados referentes a la investigación motivo de esta tesis doctoral:

- Hurtado, Omar. Patrones de requisitos: aplicación de técnicas de reúso en la ingeniería de requisitos. 3° Congreso Internacional de Ingeniería, organizado por la Universidad de Cundinamarca (UDEC) – Colombia. 2 y 3 de octubre de 2014.
- Hurtado, Omar. Reúso de activos software. IX Congreso Internacional de Ingeniería de Software, Sistemas de Información y Telecomunicaciones. Universidad Privada Antenor Orrego (UPAO) – Trujillo – Perú. Del 12 al 19 de noviembre de 2012.
- Hurtado, Omar. Reúso de activos software: aplicación al ámbito de la ingeniería de requisitos. XII Congreso Nacional de Ingeniería Industrial y de Sistemas. “Hacia un desarrollo sostenible con tecnología y creatividad competitiva y responsable”. Auditorio del Colegio de Ingenieros del Perú – Piura – Perú (CIP). Del 1 al 5 de diciembre de 2009.
- Hurtado, Omar. Técnicas de reúso de activos software y su aplicación a activos de ingeniería de requisitos. Evento científico tecnológico. Organizado por el Fondo para la Innovación, Ciencia y Tecnología (FINCyT) del Perú y la Fundación Universitaria Iberoamericana (FUNIBER) en Salón Atlantis de los Delfines Hotel & Casino – Lima - Perú. 19 de noviembre de 2009.
- Hurtado, Omar. Servicio web para reúso de requisitos, basado en el modelo de requisitos orientado al reúso efectivo (WEBMORORE). I Congreso Internacional de Ingeniería. Universidad Cesar Vallejo de Trujillo – Perú. Del 17 al 20 de noviembre del 2010.
- Hurtado, Omar. El reúso automático de requisitos en el proceso de desarrollo de software. Universidad Particular de Iquitos (Perú), Facultad de ingeniería. 15 de Julio de 2006.

### 5.4.3 Proyectos I+D

A continuación, listamos los proyectos donde se emplearon parte de las técnicas descritas en la presente tesis doctoral:

- Desarrollo de un sistema de recuperación conceptual mediante niveles semánticos en la representación de esquemas metadatos. Ministerio de Educación y Ciencia de España y Universidad Carlos III de Madrid. Referencia: TIN2007-67153. Del 01/10/2007 al 30/09/2010.

## CONCLUSIONES

---

- Combinación de técnicas no convencionales aplicadas a la extracción de información. Comunidad de Madrid y Universidad Carlos III de Madrid. Referencia: 2008/00086/001. Del 01/01/2008 al 31/12/2008.
- Proyecto VARAYOC. Modernización de los sistemas de comando y control de los buques de la Marina de Guerra del Perú. Universidad de Piura, Marina de Guerra del Perú. Del 21 de enero al 4 de marzo de 2006.
- Sistema de representación y recuperación semántica de esquemas basados en ontologías multinivel. Comunidad de Madrid y Universidad Carlos III de Madrid. Referencia: UC3M-INF-05-052. Del 01/01/2006 al 31/12/2006.

### 5.4.4 Proyectos fin de carrera

Asimismo, tenemos proyectos fin de carrera, directamente relacionados con la presente tesis doctoral. Estos proyectos ideados por el autor de la presente tesis doctoral los listamos a continuación:

- Gestor avanzado de requisitos y patrones de atributos de requisitos. Proyecto de fin de carrera. María Marchena Ronda. Universidad Carlos III de Madrid. Titulación: Ingeniería Técnica en Informática de Gestión (2009).
- Clasificación de tipos de requisitos para la mejora del proceso de desarrollo de software. Proyecto fin de Carrera. Adelaida Ramírez Fernández. Universidad Carlos III de Madrid. Titulación: Ingeniería en Informática (2012).
- Metodología para el reúso efectivo de patrones de requisitos en la ingeniería de software. Proyecto fin de carrera. Eduardo Cáceres Saldaña. Universidad de Piura. Titulación: Ingeniería Industrial y de Sistemas (2014).
- Modelo para medir calidad en patrones de requisitos y estructuras de tipos para reúso efectivo. Proyecto fin de carrera. Milagritos Carrión Vilchez. Universidad de Piura. Titulación: Ingeniería Industrial y de Sistemas (2011).
- Aplicación UML para representar una propuesta de reúso de requisitos. Proyecto fin de carrera. Kimberly Sánchez Holguín. Universidad de Piura. Titulación: Ingeniería Industrial y de Sistemas (2016).

### 5.4.5 Aplicaciones

- Parte de la propuesta de presente tesis doctoral se implementó en una herramienta CASE (swREUSER. The Reuse Company. Available: <http://www.reusecompany.com> [última visita 15-06-2005]). Específicamente el módulo de patrones de requisitos basado en las ideas de MORORE (Específicamente el Modelo de Patrones de Requisitos). Se comercializó por la empresa The Reuse Company la herramienta swREUSER. Este módulo implantado en swREUSER se encuentra en el menú principal de Life Cycle, dentro del submenú Requirements y dentro del submenú Requirements Patterns.
- Otra aplicación, por ahora en proceso de desarrollo, es WEBMORORE. Aplicación que está dentro del marco de un proyecto de fin de carrera en la Universidad de Piura. Esta aplicación permite crear patrones de requisitos con su respectiva descripción (conjunto de atributos descriptivos del patrón) y el conjunto de requisitos constitutivos del patrón.

## CONCLUSIONES

---

### 5.4.6 Universidades

- Actualmente la herramienta swREUSER, con el módulo de patrones de requisitos, es usada para la enseñanza de Facultad de Ingeniería de la Universidad de Piura – Perú. Específicamente en la asignatura de “Análisis y Diseño de Sistemas” del programa académico de “Ingeniería Industrial y de Sistemas”.

## REFERENCIAS

### 6 Referencias

(Abushark, et al., 2016) Abushark, Y. Thangarajah, J. Miller, T. Winikoff, M. Harland, J. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. ISBN: 978-1-4503-4239-1. (2016).

(Alexander, et al., 1977) Alexander, C. Ishikawa, S. Silverstein, M. Jacobson, M. Fiksdahl-King, I. Angel, S. A Pattern Language: Towns, Buildings, Construction. Oxford University Press. ISBN: 0195019199. (1977).

(Angulo, 2007) Angulo, César. Estadística. 2ed. Universidad de Piura. ISBN: 9789972481154. (2007).

(Baeza-Yates & Ribeiro-Neto, 1999) Baeza-Yates, R. Ribeiro-Neto, B. Modern Information Retrieval. Addison-Wesley. ISBN: 020139829X. (1999).

(Bassett, 1987) Bassett, P.G. Frame-Based Software Engineering. In IEEE Software, vol. 4(4), pp 9-16. ISSN: 07407459. (1987).

(Basili & Musa, 1991) Basili, V. Musa, J. The Future Engineering of Software: A Management Perspective. In Computer, vol. 24(9), pp. 90-96. Doi:10.1109/2.84903. (1991).

(Beltran, 2005) Beltran, O. Revisiones sistemáticas de la literatura. In Revista Colombiana de Gastroenterología, vol. 20(1), pp. 60-69. ISSN: 0120-9957 (2005).

(Boehm, 1993) Boehm, B. Economic Analysis of Software Technology Investments. Analytical Methods in Software Engineering Economics of Thomas Guldedge and William Hutzler, eds. Springer-Verlag. ISBN: 0387559507. (1993).

(Boehm, 1999) Boehm, B. Managing Software Productivity and Reuse. In IEEE Computer, vol. 32(9), pp. 111–113. ISBN: 00189162. (1999).

(Braude & Bernstein, 2016) Braude, J. Bernstein, M. Software Engineering: Modern Approaches. Waveland Press. ISBN: 1478632305. (2016).

(Brooks, 1987) Brooks, F. No Silver Bullet - Essence and Accidents of Software Engineering. In IEEE Computer, vol. 20(4), pp. 10-19. ISSN: 00189162. (1987).

(Burccella, et al., 2013) Buccella, A. Luzuriaga, J. M. Cechich, A. Martínez, R. Mazalu, R. Cruz, M. Pol'la, M. Arias, M. Martín, A. Doldan, M. Morsán, E. Reúso de Software Orientado a Dominios. In SEDICI Repositorio Institucional de la UNLP: “XV Workshop de Investigadores en Ciencias de la Computación”. (2013).

(Burch, 2005) Burch, S. Sociedad de la información/Sociedad del conocimiento. In Ambrossi, A.; Peugeot, V.; Pimienta, D. Palabras en juego: Enfoques Multiculturales sobre las Sociedades de la Información. París: C & F Editions. ISBN: 2-915825-03-3. (2005).

(Buschmann, et al., 2007) Buschmann, F. Henney, K. Schmidt, D. Pattern-Oriented Software Architecture Volume 5. On Patterns and Pattern Languages. A System of Patterns. Wiley. ISBN: 0471486485. (2007).

## REFERENCIAS

---

(Büttcher, et al., 2010) Büttcher, S. Clarke, C. Cormack, G. Information Retrieval Implementing and Evaluating Search Engines. The MIT Press. ISBN: 978-0-262-02651-2. (2010).

(Cáceres, 2014) Cáceres, E. Hurtado, O. (Director). Metodología para el reúso de patrones de requisitos en la ingeniería de software. Proyecto de fin de carrera. Universidad de Piura. Programa académico de Ingeniería Industrial y de Sistemas. (2014).

(Campos, et al., 2015) Campos, R. Dias, G. Jorge, A. Jatowt, A. Survey of Temporal Information Retrieval and Related Applications. In Journal ACM Computing Surveys (CSUR). Vol. 47(2). Doi: 10.1145/2619088. (2015).

(CC, 2017) The Common Criteria Recognition Agreement. Common Criteria for Information Technology Security Evaluation. Available: <http://www.commoncriteriaportal.org/>. [last visited on 03-01-2017].

(Chavarriaga & Arboleda, 2004) Chavarriaga, J. Arboleda, H. Modelo de Investigación en Ingeniería de Software: Una propuesta de investigación tecnológica. In Proceedings del 2º Workshop en Métodos de Investigación y Fundamentos Filosóficos en Ingeniería de Software y Sistemas de Información del IV Congreso de la Sociedad de Lógica, Metodología y Filosofía de la Ciencia de España. (2004).

(Christel & Kang, 1992) Christel, M. & Kang, K. Issues in Requirements Elicitation. Carnegie Mellon Software Engineering Institute. Technical Report. CMU/SEI-92-TR-12. (1992).

(Cobarsí-Morales, et al., 2013) Cobarsí-Morales, J. Canals, A. Ortoll, E. La información como bien económico: reflexiones sobre as crisis financiera 2008. In El Profesional de la Información. Vol. 22(4), pp 346-352. ISSN: 1386-6710.

(Coplien, 1992) Coplien, J. Advanced C++ Programming Styles and Idioms. Addison-Wesley. ISBN: 0201548550. (1992).

(CORE, 2017) Vitech: Insight Through Integration. CORE. Available: <http://www.vitechcorp.com/> [last visited on 05/01/2017].

(Crosby, 1996) Crosby, P. Quality is still free: making quality certain in uncertain times. McGraw-Hill. ISBN: 0070145326. (1996).

(Croft & Lafferty, 2003) Croft, B. Lafferty, J. Language Modeling for Information Retrieval. Springer. ISBN: 1402012160. (2003).

(Cunningham & Kent, 1988) Cunningham, W. Kent, B. Using a pattern language for programming. In Addendum to the Proceedings of OOPSLA'87, of ACM SIGPLAN Notices, vol. 23(5), pp 16. ISSN: 0362-1340. (1988).

(Cybulski & Reed, 2000) Cybulski, J. Reed, K. Requirements Classification and Reuse: Crossing Domain Boundaries. In proceedings 6th International Conference, ICSR-6, Vienna, Austria, June 27-29. (2000) and in Lecture Notes in Computer Science Vol. 1844, pp 190-210. ISBN: 978-3-540-67696-6. (2000).

## REFERENCIAS

---

(Deming, 1989) Deming E. Calidad, productividad y competitividad. La salida de la crisis. Ediciones Díaz de Santos. ISBN: 8487189229. (1989).

(DOORS, 2017) IBM. Rational DOORS. Available: <http://www-03.ibm.com/software/products/es/ratidoor>.

(Durán, et al., 2000) Durán, A., Ruiz, A., Corchuelo, R. y Toro, M. Identificación de patrones de reutilización de requisitos de sistemas de información. In Proceedings of III Workshop em Engenharia de Requisitos WER'2000, pp. 230-242. (2000).

(Durán, et al., 1999) Durán, A. Bernárdez, B. Ruiz, A. Toro, M. A Requirements Elicitation Approach Based in Templates and Patterns. In proceedings 2<sup>nd</sup>. Workshop on Requirements Engineering (WER 99). Argentina. (1999).

(Encuestafacil, 2017) Encuestafacil. Encuestafacil. Available: <https://www.encuestafacil.com/>. [last visited on 10-01-2017].

(ESA, 2017) European Space Agency. ESA Software Engineering & Standardisation: ESA PSS-05-02 Issue 1 Revision 1: Guide to the user requirements definition phase. And, ESA PSS-05-03 Issue 1 Revision 1: Guide to the software requirements definition phase Available: [http://www.esa.int/TEC/Software\\_engineering\\_and\\_standardisation/](http://www.esa.int/TEC/Software_engineering_and_standardisation/). [last visited on 17-02-2017].

(Fayad & Flood, 2016) Fayad, M. Flood, C. Unified Software Engineering Reuse (USER) using stable analysis, design and architectural patterns. In 2016 Future Technologies Conference (FTC). Doi: 10.1109/FTC.2016.7821682. (2016).

(Fayad, et al., 2014) Fayad, M. Sanchez, H. Hegde, S. Basia, A. Vakil, A. Software Patterns, Knowledge Maps, and Domain Analysis. Auerbach Publications. ISBN: 9781466571433. (2014).

(Fernández-Medina, 2003) Fernández-Medina, E. Moya, R. Piattini, M. Seguridad de las Tecnologías de Información: la construcción de la confianza para una sociedad conectada. Ediciones AENOR. ISBN: 9788481433678. (2003).

(Fowler, 2017) Fowler, M. WEBSITE of Martin Fowler. Available: <http://martinfowler.com/>. [last visited on 15-01-2017].

(Fowler, 1997) Fowler, M. Analysis Patterns: Reusable Object Models. Addison-Wesley Professional. ISBN: 0201895420. (1997).

(Fraga, 2010) Fraga, A. A Methodology for Reusing Any Kind of Knowledge at Low Cost: Universal Knowledge Reuse. Tesis doctoral. Universidad Carlos III de Madrid, Departamento de Ingeniería Informática. (2010).

(Fuentes, et al., 2016) Fuentes, J. Fraga, A. Génova, G. Parra, E. Alvarez, J. Llorens, J. Applying INCOSE Rules for writing high-quality requirements in Industry. In INCOSE International Symposium. Vol. 26(1), pp. 1875-1889. doi:10.1002/j.2334-5837.2016.00267.x. (2016).

## REFERENCIAS

---

(Gamma, et al., 1995) Gamma, E. Helm, R. Jonson, R. & Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN: 0201633612. (1995).

(Garshol, 2004) Garshol L.M. Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. Journal of information science. Vol. 30(4), pp. 378-91. (2004).

(Garvin, 1984) Garvin, D. What Does 'Product Quality' Really Mean? In MIT Sloan Management Review. vol 26 (1), pp. 25-45. (1984).

(Génova, et al., 2017) Génova, G. Hernanz, I. Galindos, R. Hurtado, O. Apuntes de la asignatura de Ingeniería de Software I, Departamento de Informática de la Universidad Carlos III de Madrid. Available: <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/Is1y2/IS1.htm>. [last visited on 13-01-2017].

(Génova, et al., 2009) Génova, G. Valiente, M. Marrero, M. On the difference between analysis and design, and why it is relevant for the interpretation of models in Model Driven Engineering. In Journal of Object Technology, vol. 8(1), pp 107-127. Doi: 10.5381/jot.2009.8.1.c7. (2009).

(Génova, et al., 2013) Génova, G. Fuentes, J. M. Llorens, J. Hurtado, O. Moreno, V. A framework to measure and improve the quality of textual requirements. In Journal of Requirement Engineering. Vol. 18(1), pp 25-41. ISSN: 0947-3602. (2013).

(Génova, et al., 2012) Génova, G. Llorens, J. Morato, J. Software engineering research: the need to strengthen and broaden the classical scientific method. In Manuel Mora, Ovsei Gelman, Annette L. Steenkamp and Mahesh Raisinghani (eds.), Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems. IGI Global pp. 106-125. (2012).

(Genero, et al., 2014) Genero, M., Cruz-Lemus, J.A., Piattini, M. Métodos de Investigación en Ingeniería del Software. RaMa. ISBN: 978-849964-507-0. (2014).

(Govindaraju & Wiradanti, 2015) Govindaraju, R. Wiradanti, B. Factors influencing the willingness to implement requirement engineering good practices. In International Journal of Business Information Systems. Vol. 19(3), pp 279-299. (2015).

(Gruber, 1993) Gruber, T. A Translation Approach to Portable Ontology Specifications. In Knowledge Acquisition, vol. 5(2), Hardcover, 4th Edition, 852 pages, McGraw-Hill Companies pp. 199-220. ISSN: 10428143. (1993).

(Guerrero, et al., 2012) Guerrero, D. Martinez-Almeda, J. La Rosa, G. Professional competences: intellectual structure of research. In J. Pantouvakis (Ed.), Proceedings of the Twenty Sixth International Project Management Association, pp. 618-625. Grecia: IPMA. (2012).

(Happel, et al., 2006) Happel, H. Korthaus, A. Seedorf, S. Tomczyk, P. KOntoR: An Ontology-enabled Approach to Software Reuse. In proceedings The 18th International Conference, Software engineering and Knowledge engineering, SEKE 2006, pp. 349-354. ISBN: 1-891706-18-7. (2006).



## REFERENCIAS

---

(Horkoff & Yu, 2016) Horkoff, J. Yu, E. Interactive goal model analysis for early requirements engineering. In Journal of Requirements Engineering. Vol. 21(1), pp. 29-61. Doi: 10.1007/s00766-014-0209-8. (2016)

(Hurtado, et al., 2005) Hurtado, O. Llorens, J. Génova, G. Fuentes, J.M. Generación de Patrones de Requisitos en una Herramienta CASE: Aplicación al Common Criteria. In Hernán Astudillo & Carla Taramasco, ed., 'CibSE', pp. 21-32. ISBN: 956-7051-07-0. (2005).

(Hurtado, et al., 2006) Hurtado, O. Fraga, A. Hernanz, I. Llorens, J. Metamodel of requirements' types classifications for improving the software development process (SDP): an ontological approach. IASTED International Conference on Software Engineering and Applications (SEA2006). Dallas (Texas), USA. ISBN Hardcopy: 0889866422 / CD: 088986599X. (2006).

(Hurtado, et al., 2009) Hurtado, O. Quinde, M. Martín, D. Particularización dinámica de plantillas para la gestión de requisitos. In Journal of Tecnología y Desarrollo. Vol. 7(1), pp 121-126. ISSN: 1819-4575. (2009).

(Ibañez & Rempp, 1996) Ibañez, M. and Rempp, H. European User Survey Analysis. European Software Institute, Report USV\_EUR, version 2.1. ESPITI Project. (1996).

(IEEE, 1990) IEEE Computer Society. IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. ISBN: 155937067-X. (1990).

(ISO/IEC/IEEE-29148, 2017) ISO/IEC/IEEE 29148:2011. Systems and software engineering -- Life cycle processes -- Requirements engineering. Available: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=45171](http://www.iso.org/iso/catalogue_detail.htm?csnumber=45171) . [last visited on 23-01-2017].

(IRQA, 2017) Visure The Requirements Company. IRQA. Available: <http://www.visuresolutions.com/>. [last visited on 20-01-2017].

(INCOSE, 2017) INCOSE. The International Council on Systems Engineering. Available: [www.incose.org/](http://www.incose.org/). [last visited on 16-01-2017].

(ISO/IEC-14598-1, 2017) International Organization for Standardization. ISO/IEC 14598-1. Available: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=24902](http://www.iso.org/iso/catalogue_detail.htm?csnumber=24902). [last visited on 08-02-2017].

(ISO/IEC-12207, 2017) International Organization for Standardization. ISO/IEC 12207:2008. Available: [http://www.iso.org/iso/catalogue\\_detail?csnumber=43447](http://www.iso.org/iso/catalogue_detail?csnumber=43447). [last visited on 08-02-2017].

(ISO-9000, 2017) International Organization for Standardization. ISO 9000 – Quality management. Available: <http://www.iso.org/iso/home.html> . [last visited on 04-01-2017].

## REFERENCIAS

---

(Ito, 2016) Ito, M. Cardion.spec: An Approach to Improve the Requirements Specification Written in the Natural Language Through the Formal Method. In proceedings 23rd European Conference, EuroSPI 2016. Vol. 633, pp. 58-69. ISBN: 978-3-319-44816-9. (2016).

(Juran & De Feo, 2010) Juran, J. De Feo, J. Juran's Quality Handbook. 6ed. McGraw Hill. ISBN: 0071629734 (2010).

(Kanhbua & Anand, 2016) Kanhbua, N. Anand, A. Temporal Information Retrieval. In proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ISBN: 978-1-4503-4069-4. (2016).

(Kaur, et al., 2013) Kaur, A. Grover, P.S. Dixit, A. An Improved Model to Estimate Quality of the Software Product. In Journal of Research YMCAUST International. Vol. 1(2), pp. 6-11. ISBN: 2319-9377. (2013).

(Kircher & Jain, 2004) Kircher, M. Jain, P. Pattern-Oriented Software Architecture Volume 3: Patterns for Resource Management. Wiley. ISBN: 978-0-470-84525-7.

(Koh, 2016) Koh S. Cause-and-Effect Perspective on Software Quality: Application to ISO/IEC 25000 Series SQuARE's Product Quality Model. In JITAM Journal of Information Technology Applications and Management. Vol. 23(3), 71-86. ISBN: 1598-6284. (2016).

(Konrad & Cheng, 2002) Konrad, S. Cheng, B. Requirements Patterns for Embedded Systems. In proceedings of the IEEE Joint International Conference on Requirements Engineering (RE02), Essen, Germany. ISBN: 0-7695-1465-0. (2002).

(Kotonya, 1998) Kontoya, G. Sommerville, I. Requirements engineering: processes and techniques. John Wiley & sons. ISBN: 0471972088. (1998).

(Krueger, 1992) Krueger, C.W. Software Reuse. In ACM Computing Surveys (CSUR), vol. 24(2), pp. 131-183. ISSN: 0360-0300. (1992).

(KR, 2017) Knowledge Reuse Group. The Knowledge Reuse Group. Available: [http://www.uc3m.es/ss/Satellite/UC3Minstitucional/es/Detalle/Organismo\\_C/1371210361342/1371211778776/Knowledge\\_Reusing](http://www.uc3m.es/ss/Satellite/UC3Minstitucional/es/Detalle/Organismo_C/1371210361342/1371211778776/Knowledge_Reusing) . [last visited on 26-01-2017].

(Laguna, et al., 2001) Laguna, M. García, F. López, O. Marqués, J. Reutilización de Requisitos en el Modelo Mecano. In I Jornadas de Ingeniería de Requisitos Aplicada (JIRA), Sevilla, pp. 111-128. (2001).

(Lamarca, 2017) Lamarca, M. J. Indización y Clasificación de Hipertextos. Available: <http://www.hipertexto.info/documentos/indizacion.htm> . [last visited on 16-02-2017].

(Laplante, 2013) Laplante, P. Requirements Engineering for Software and Systems. 2ed. Auerbach Publications. ISBN: 9781466560819. (2013).

## REFERENCIAS

---

(Larman, 2004) Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3ed. Prentice Hall. ISBN: 0131489062. (2004).

(Llorens, 1996) Llorens, J. Definición de una metodología y una estructura de repositorio orientadas a la reutilización: el tesoro de software. Tesis doctoral. Universidad Carlos III de Madrid, Departamento de Ingeniería. (1996).

(Loucopoulos & Karakostas, 1995) Loucopoulos, P. & Karakostas, V. Systems Requirements Engineering. McGraw-Hill International series in Software Engineering. ISBN: 0077078438. (1995).

(Manning, et al., 2008) Manning, C. Raghavan, P. Schütze, H. Introduction to Information Retrieval. Cambridge University Press. ISBN: 0521865719 (2008).

(Manoel, et al., 2013) Manoel, M. Lencastre, M. Araújo, J. Abordagem para Reuso de Requisitos Tardios em Sistemas de Informação. In Proceedings of Requirements Engineering@Brazil 2013. Vol. 1005, pp 43-49. ISSN: 1613-0073. (2013).

(Marrero, et al., 2008) Marrero, M. Sánchez-Cuadrado, S. Fraga, A. Llorens, J. Applying Ontologies and Intelligent Text Processing in Requirements Reuse. In First Workshop on Knowledge Reuse (KREUSE'08). In 10th International Conference on Software Reuse (ICSR'08), pp. 25–29. Beijing, China. ISBN: 9788469131664. (2008).

(Manber & Myers, 1990) Manber, U. Myers, G. Suffix arrays: a new method for on-line string searches. On proceedings SODA'90: Proc. of the Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 319–327. (1990).

(McIlroy, 1968) McIlroy, M. Mass produced software components. In P. Naur and B. Randell, eds. Software engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany. Scientific Affairs Division, NATO, Brussels, 1969, pp. 138-155. (1968).

(Metrica-3, 2017) Ministerio de Administraciones Públicas de España. Métrica Versión 3. Available: [http://administracionelectronica.gob.es/pae/Home/pae\\_Documentacion/pae\\_Metodologia/pae\\_Metrica\\_v3.html#U\\_NV3WPCfnF020139815X](http://administracionelectronica.gob.es/pae/Home/pae_Documentacion/pae_Metodologia/pae_Metrica_v3.html#U_NV3WPCfnF020139815X) [last visited on 17-01-2017].

(Mili, et al., 2002) Mili, H. Mili, A. Yacoub, S. y Addy, E. Reuse Based Software Engineering: Techniques, Organization, and Measurement. John Wiley & Sons, INC. ISBN: 0471398195. (2002).

(Mili, et al., 1997) Mili, H. Ah-Ki, E. Godin, R. Mccheick, H. Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval. Software Engineering Notes, vol. 22(3), pp. 89–98. ISBN: 0-89791-945-9. (1997).

(Moros, et al., 2008) Moros, B. Vicente-Chicote, C. Toval, A. REMM-Studio + : Modeling Variability to Enable Requirements Reuse. In Proceedings 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20-24, 2008. Vol. 5231, pp. 530-531. ISBN: 978-3-540-87876-6. (2008).

## REFERENCIAS

---

(Neighbors, 1984) Neighbors, J. The Draco approach to constructing software from reusable components. In IEEE Transactions on Software Engineering, SE, vol. 10(5), pp. 564–574. ISSN: 0098-5589. (1984).

(Nicolás, et al., 2003) Nicolás J. Lasheras J. Toval A. y Moros B. Soporte automatizado a la reutilización de requisitos. In VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'03), pp. 335-346. ISBN: 8468838365. (2003).

(NTP, 2017) INDECOPI. Norma Técnica Peruana: “NTP-ISO/IEC 12207:2006 TECNOLOGIA DE LA INFORMACIÓN. Procesos del ciclo de vida de software. 1ª Edición”. Available: [http://www.senasa.gob.pe/senasa/wp-content/uploads/2014/11/Certificacion-citricos-a-mexico\\_26\\_mayo\\_2105\\_2.pdf](http://www.senasa.gob.pe/senasa/wp-content/uploads/2014/11/Certificacion-citricos-a-mexico_26_mayo_2105_2.pdf). [last visited on 27-01-2017].

(Nystrom, 2014) Nystrom, R. Game Programming Patterns. Genever Benning. ISBN: 0990582906. (2014).

(Orange-Book, 2017) Department of Defense. Trusted Computer System Evaluation Criteria: DOD 5200.28-STD. Available: <http://www.fas.org/irp/nsa/rainbow/std001.htm>. [last visited on 27-01-2017].

(Pacheco, 2016) Pacheco, C. Garcia, I. Calvo-Manzano, J. A. Arcilla, M. Reusing functional software requirements in small-sized software enterprises: a model oriented to the catalog of requirements. In Requirements Engineering. pp. 1-13. Doi: 10.1007/s00766-015-0243-1. (2016).

(Parra, 2016) Parra, E. Metodología orientada a la optimización automática de la calidad de los requisitos. Tesis Doctoral. Universidad Carlos III de Madrid. (2016).

(Peña, 2010) Peña, D. Regresión y Diseño de Experimentos. Edición 2010. Alianza Editorial. ISBN: 9788420693897. (2010).

(PMI, 2017) PMI. Project Management Institute. Available: <http://www.pmi.org/> [last visited on 11-01-2017].

(PMBok, 2013) PMI. A Guide to de the Project Management Body of Knowledge. Fifth Edition. By PMI Publications. ISBN: 978-1-935589-67-9. (2013).

(Pressman & Maxim, 2015) Pressman, R. Maxim, B. Software Engineering: a practitioner's approach. 8ed. McGraw-Hill Education. ISBN: 0078022126. (2016).

(Pressman, 2002) Pressman R. Ingeniería de Software: un Enfoque Práctico. 5ed. McGraw-Hill/Interamericana de España S.A.U. ISBN: 8448132149. (2002).

(Prieto-Diaz, 1987) Prieto-Diaz, R. Freeman, P. Software reusability. In Software IEEE Computer Society Press, vol. 4(1), pp 6-16. ISSN: 07407459. (1987).

(Rajani & Rama, 2016) Rajani, T. Rama, B. Benefits of Software Reuse. In proceedings International Conference On Futuristic Trends in Engineering, Science, Humanities, and Technology FTESHT-16. Vol. 2(3), pp. 10-14. ISBN: 978-93-85225-55-0. (2016).

## REFERENCIAS

---

(RDD-100, 2017) ALC: Volere Requirements Resources. RDD-100. Available: <http://www.volere.co.uk/tools/rdd-100-version-4-1-2> . [last visited on 22-10-2017].

(Reqstudio, 2017) The Knowledge Reuse Group. Requirements Studio. Available: <https://reqstud.wordpress.com/> . [last visited on 18-01-2017].

(Reppro, 2017) Rational. Rational Requisite Pro. Available: <http://www-01.ibm.com/software/awdtools/reqpro/>. [last visited on 22-01-2017].

(Rini & Govilkar, 2016) Rini, J. Govilkar, S. Survey of Information Retrieval Techniques for Web using NLP. In International Journal of Computer Applications. Vol. 135(8), pp. 23-27. (2016).

(Robertson & Robertson, 2012) Robertson, S. Robertson, J. Mastering the Requirements Process: Getting Requirements Right. Addison-Wesley. ISBN: 0321815742. (2012).

(Rosique, et al., 2014) Rosique, M. Cáceres, D. Sánchez, P. Sánchez, F. Reúso efectivo de modelos domóticos a través de requisitos genéricos. In Anuario de Jóvenes Investigadores. Vol. 7, pp. 60-62. ISSN: 2386-3676. (2014).

(Saavedra, et al., 2013) Saavedra, R. Ballejos, L. Ale, M. Software Requirements Quality Evaluation: State of the art and research challenges. In 14th Argentine Symposium on Software Engineering, ASSE 2013. Pp. 240-257. ISSN: 1850-2792. (2013).

(Saeed, et al., 2016) Saeed, M. Sarwar, N. Bilal, M. Efficient requirement engineering for small scale project by using UML. In proceedings of 2016 Sixth International Conference on Innovative Computing Technology (INTECH). Pp. 662-666. Doi: 10.1109/INTECH.2016.7845116. (2016).

(Salazar, et al., 2015) Salazar, D., Hurtado, O, Génova, G. Validación de la aplicación de un modelo para la mejora de la ingeniería de requisitos. En actas del 1° Congreso Internacional de Ingeniería: Aporte de la ingeniería al desarrollo sostenible del Perú. Universidad Privada Antenor Orrego de Trujillo. (2015).

(Sánchez, 2016) Sánchez, K. Hurtado, O. (Director). Aplicación de UML para representar una propuesta de reúso de requisitos. Proyecto de fin de carrera. Universidad de Piura. Programa académico de Ingeniería Industrial y de Sistemas. (2016).

(Sánchez, et al., 2015) Sánchez, K, Hurtado, O, Génova, G. Modelo para medir la calidad de activos de la ingeniería de requisitos. En actas del 1° Congreso Internacional de Ingeniería: Aporte de la ingeniería al desarrollo sostenible del Perú. Universidad Privada Antenor Orrego de Trujillo. (2015).

(Sankhwar, et al., 2014) Sankhwar, S. Virendra, S. Dhirendra, P. Requirement engineering paradigm. In Global Journal of Multidisciplinary Studies. Vol 3(3). (2014).

(Schweiger, 2013) Schweiger, Andreas. Applying software patterns to requirements engineering for avionics systems. In Systems Conference (SysCon), 2013 IEEE International. DOI: 10.1109/SysCon.2013.6549853. (2013).

## REFERENCIAS

---

(SCRIBD, 2017) Scribd. Chaos Manifiesto 2013. Available: <https://es.scribd.com/document/198550543/Chaos-Manifiesto-2013>. [last visited on 16-01-2017].

(Seco, 2009) Seco, D. Técnicas de Indexación y Recuperación de documentos utilizando referencias geográficas y textuales. Tesis Doctoral. Universidad de la Coruña, Departamento de Computación. (2009).

(SERENA, 2017) SERENA. Serena RTM. Available: <http://www.serena.com>. [last visited on 27-01-2017].

(Shah & Patel, 2014) Shah, T. Patel, V. A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods. In International Journal of Computer Applications. Vol. 99(15), pp. 36-45. (2014).

(SIIU, 2010) Ministerio de Educación, Cultura y Deporte de España. Sistema Integrado de Información Universitaria - Especificación de Requisitos. (2010).

(Sommerville, 2016) Sommerville I. Ingeniería de Software. 10ed. Pearson Educación, S.A. ISBN: 1292096136. (2016).

(Sommerville, 2001) Somerville, I. Software. Engineering 6th Edition. Addison Wesley. ISBN: 020139815X. (2001).

(Sommerville & Sawyer, 1999) Sommerville, I. Sawyer, P. Requirements engineering: a good practice guide. John Wiley & Sons. ISBN: 0471974447. (1999).

(Srivastava, 2013) Srivastava, S. A Repository of Software Requirement Patterns for Online Examination System. In IJCSI International Journal of Computer Science Issues Vol. 10, Issue 3, No 2. ISSN: 1694-0814. (2013).

(Standish, 2016) The Standish Group. Chaos Report. Available: <http://www.standishgroup.com/>. [last visited on 17-08-2016].

(Statgraphics, 2017) Statgraphics. Statgraphics. Available: <http://www.statgraphics.com/>. [last visited on 7-01-2017].

(Suárez & Gutierrez, 2016) Suárez, J. Gutierrez, L. Tipificación de Dominios de Requerimientos para la Aplicación de Patrones Arquitectónicos. In Journal Información Tecnológica. Vol. 27(4). ISSN: 0718-0764. (2016).

(Sutcliffe, et al., 2006) Sutcliffe, A. Papamargaritis, G. Zhao, L. Comparing requirements analysis methods for developing reusable component libraries. In Journal of Systems and Software, vol. 79, pp.273-289. DOI: 10.1016/j.jss.2005.06.027. (2006).

(SWEBOK, 2017) SWEBOK. Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, 2004. Available: <http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf>. [last visited on 17-01-2017].

(swReuser, 2017) The Reuse Company. swREUSER. Available: <http://dtinf-the-reuse-company.software.informer.com/> [last visited on 30-01-2017].



## REFERENCIAS

---

(Thayer & Dorfman, 1997) Thayer, R. & Dorfman, M. Software Requirements Engineering. 2ed. IEEE Computer Society Publications. ISBN: 0818677384. (1997).

(Toval, et al., 2002) Toval, A. Nicolás, J. y Moros, B. SIREN: Un Proceso de Ingeniería de Requisitos Basado en Reutilización. In cap. 4, in Applying Requirements Engineering, Catedral Publicaciones, pp. 57-71. ISBN: 8496086062. (2002).

(TRC, 2017) The Reuse Company. Presente y futuro de la Reutilización de Software. Available: <https://www.adictosaltrabajo.com/tutoriales/reuse/>. [last visited on 30-01-2017].

(Trejo, 2016) Trejo, R. Viviendo en el Aleph. La sociedad de la información y sus laberintos. Gedisa. ISBN: 849784131X. (2016).

(UML, 2017) Object Management Group. Unified Modeling Language. Available: <http://www.uml.org/>. [last visited on 16-01-2017].

(Vavassori & Cezario, 2013) Vavassori, F. Cezario, R. Evaluation of a Systematic Approach to Requirements Reuse. In Journal of Universal Computer Science. Vol. 19(2), pp. 254-280. (2013).

(Withall, 2010) Withall, S. Software Requirement Patterns. 1 Review. Microsoft Press. ISBN: 9780735646063. (2010).

(Withall, 2014) Withall, S. WEBSITE of Stephen Withall. Available: <http://www.withallyourequire.com/aboutsteve.html>. [last visited on 02/08/2014]

(Zapata, et al., 2006) Zapata, C. Jaramillo, A. Arango, F. Una propuesta para mejorar la completitud de requisitos utilizando un enfoque lingüístico. In Journal Ingeniería & Desarrollo. Vol. 19, pp. 1-16. (2006).

(Zave, 1997) Zave, P. Classification of Research Efforts in Requirements Engineering. In ACM Computing Surveys (CSUR), vol. 29(4), pp. 315-321. ISSN: 03600300. (1997).

(Zobel & Moffat, 2006) Zobel, J. Moffat, A. Inverted files for text search engines. ACM Computing. Surveys. Vol 38(2):6. (2006).

# ANEXOS



## 7 Anexos

### 7.1 Anexo 1: grupos de alumnos con calificaciones

Sede	1. Piura
	2. Lima
	3. ETS
Herramienta	1. Microsoft Word
	2. Software REUSE Studio – Herramienta Case
	3. Software REUSE Studio con Patrones
Salón	1. ETS - 2013 II
	2. ADS - 2012 I
	3. ADS - 2009 I
	4. ADS - 2013 I (Lima)
	5. ADS - 2014 I (Lima)
	6. ETS - 2012 I
	7. ADS - 2011 I
	8. ADS - 2010 I
	9. ETS - 2013 I
	10. ETS - 2014 I
	11. ADS - 2013 I
	12. ADS - 2009 II (Lima)

#### 7.1.1 Data de alumnos del curso “Análisis y Diseño de Sistemas (ADS)” sede Piura.

Alumno	Nota Requisito	Sede	Herramienta	Índice académico	Tamaño de grupo	Salón
1	8	1	1	13.18	5	2
2	8	1	1	10.67	5	2
3	8	1	1	12.77	5	2
4	8	1	1	10.04	5	2
5	8	1	1	10.12	5	2
6	8	1	1	12.26	4	2
7	8	1	1	12.2	4	2
8	8	1	1	11.8	4	2
9	8	1	1	11.48	4	2
10	12	1	1	13.46	4	2
11	12	1	1	15.07	4	2
12	12	1	1	15.24	4	2
13	12	1	1	12.27	4	2
14	10	1	1	13.4	5	2
15	10	1	1	12.46	5	2
16	10	1	1	13.49	5	2
17	10	1	1	12.08	5	2
18	10	1	1	13.43	5	2
19	7	1	1	11.74	5	2

## ANEXOS

20	7	1	1	9.96	5	2
21	7	1	1	10.29	5	2
22	7	1	1	10.64	5	2
23	7	1	1	10.69	5	2
24	13	1	1	12.42	5	2
25	13	1	1	14.41	5	2
26	13	1	1	12.96	5	2
27	13	1	1	12.53	5	2
28	13	1	1	15.98	5	2
29	10	1	1	9.62	3	2
30	10	1	1	13.04	3	2
31	10	1	1	11.9	3	2
32	12	1	1	11.09	5	2
33	12	1	1	12.56	5	2
34	12	1	1	17.04	5	2
35	12	1	1	12.55	5	2
36	12	1	1	12.63	5	2
37	8	1	1	12.55	2	2
38	8	1	1	10.96	2	2
39	10	1	1	10.75	4	2
40	10	1	1	10.99	4	2
41	10	1	1	11.59	4	2
42	10	1	1	9.86	4	2
43	8	1	1	11.63	3	2
44	8	1	1	11.57	3	2
45	8	1	1	11.62	3	2
46	12	1	1	11.71	5	2
47	12	1	1	13.38	5	2
48	12	1	1	11.93	5	2
49	12	1	1	11.43	5	2
50	12	1	1	11.77	5	2
51	12	1	1	11.58	3	2
52	12	1	1	11.36	3	2
53	12	1	1	11.37	3	2
54	12	1	1	10.53	1	2
55	10	1	3	14.4	5	11
56	10	1	3	13.71	5	11
57	10	1	3	14.72	5	11
58	10	1	3	15.05	5	11
59	10	1	3	11.64	5	11
60	14	1	3	11.05	4	11
61	14	1	3	12.14	4	11
62	14	1	3	12.23	4	11
63	14	1	3	11.96	4	11

## ANEXOS

64	12	1	3	13.13	5	11
65	12	1	3	13.6	5	11
66	12	1	3	14.16	5	11
67	12	1	3	12.61	5	11
68	12	1	3	12.77	5	11
69	13	1	3	12.83	5	11
70	13	1	3	11.95	5	11
71	13	1	3	12.24	5	11
72	13	1	3	12.02	5	11
73	13	1	3	9.96	5	11
74	14	1	3	12.57	3	11
75	14	1	3	10.87	3	11
76	14	1	3	10.15	3	11
77	18	1	3	12.43	5	11
78	18	1	3	12.62	5	11
79	18	1	3	10.7	5	11
80	18	1	3	15.59	5	11
81	18	1	3	11.7	5	11
82	13	1	3	13.89	5	11
83	13	1	3	10.55	5	11
84	13	1	3	11.75	5	11
85	13	1	3	12.54	5	11
86	13	1	3	12.2	5	11
87	12	1	3	11.58	4	11
88	12	1	3	14.51	4	11
89	12	1	3	10.06	4	11
90	12	1	3	12.14	4	11
91	13	1	3	11.07	5	11
92	13	1	3	15.66	5	11
93	13	1	3	11.56	5	11
94	13	1	3	14.49	5	11
95	13	1	3	15.02	5	11
96	14	1	2	14.88	5	7
97	14	1	2	12.59	5	7
98	14	1	2	12.01	5	7
99	14	1	2	8.76	5	7
100	14	1	2	13.78	5	7
101	13	1	2	14.4	5	7
102	13	1	2	9.85	5	7
103	13	1	2	11.1	5	7
104	13	1	2	13.47	5	7
105	13	1	2	14.42	5	7
106	19	1	2	15.86	5	7
107	19	1	2	13.37	5	7

## ANEXOS

108	19	1	2	13.07	5	7
109	19	1	2	15.46	5	7
110	19	1	2	13.75	5	7
111	13	1	2	11.21	5	7
112	13	1	2	13.09	5	7
113	13	1	2	14.51	5	7
114	13	1	2	13.21	5	7
115	13	1	2	12.87	5	7
116	12	1	2	12.69	4	7
117	12	1	2	13.38	4	7
118	12	1	2	11.28	4	7
119	12	1	2	12.03	4	7
120	13	1	2	14.16	5	7
121	13	1	2	14.62	5	7
122	13	1	2	12.83	5	7
123	13	1	2	15.34	5	7
124	13	1	2	10.41	5	7
125	7	1	2	11.64	5	7
126	7	1	2	10.06	5	7
127	7	1	2	11.87	5	7
128	7	1	2	9.92	5	7
129	7	1	2	12.09	5	7
130	11	1	2	10.92	4	7
131	11	1	2	13.39	4	7
132	11	1	2	8	4	7
133	11	1	2	11.08	4	7
134	11	1	2	11.31	4	7
135	11	1	2	11.98	4	7
136	11	1	2	10.33	4	7
137	11	1	2	11.24	4	7
138	11	1	2	15.54	5	7
139	11	1	2	14.77	5	7
140	11	1	2	15.29	5	7
141	11	1	2	12.75	5	7
142	11	1	2	12.15	5	7
143	15	1	2	13.58	5	7
144	15	1	2	11.73	5	7
145	15	1	2	17.08	5	7
146	15	1	2	18.17	5	7
147	15	1	2	14.23	5	7
148	5	1	2	11.05	5	7
149	5	1	2	10.01	5	7
150	5	1	2	10.71	5	7
151	5	1	2	10.84	5	7

## ANEXOS

152	5	1	2	11.87	5	7
153	12	1	2	12.06	3	7
154	12	1	2	12.34	3	7
155	12	1	2	11.85	3	7
156	8	1	1	9.08	4	3
157	8	1	1	11.64	4	3
158	8	1	1	11.07	4	3
159	8	1	1	11.08	4	3
160	9	1	1	10.71	4	3
161	9	1	1	12	4	3
162	9	1	1	11.18	4	3
163	9	1	1	11.74	4	3
164	8	1	1	11.97	3	3
165	8	1	1	10.64	3	3
166	8	1	1	10.98	3	3
167	15	1	1	7.81	3	3
168	15	1	1	10.81	3	3
169	15	1	1	10.1	3	3
170	15	1	2	12.56	4	8
171	15	1	2	9.88	4	8
172	15	1	2	11.81	4	8
173	15	1	2	12.84	4	8
174	15	1	2	12.45	4	8
175	15	1	2	14.09	4	8
176	15	1	2	12.39	4	8
177	15	1	2	12.07	4	8
178	15	1	2	13.19	4	8
179	15	1	2	11.81	4	8
180	15	1	2	9.81	4	8
181	15	1	2	10.75	4	8
182	13	1	2	11.59	4	8
183	13	1	2	11.91	4	8
184	13	1	2	11.54	4	8
185	13	1	2	13.19	4	8
186	11	1	2	11.87	4	8
187	11	1	2	11.48	4	8
188	11	1	2	11.05	4	8
189	11	1	2	12.86	4	8
190	13	1	2	11.45	4	8
191	13	1	2	15.13	4	8
192	13	1	2	11.48	4	8
193	13	1	2	10.64	4	8

## ANEXOS

### 7.1.2 Data de alumnos del curso “Análisis y Diseño de Sistemas (ADS)” sede Lima.

Alumno	Nota Requisito	Sede	Herramienta	Índice académico	Tamaño de grupo	Salón
194	16	2	3	13.08	4	12
195	16	2	3	11.68	4	12
196	16	2	3	11.56	4	12
197	16	2	3	11.78	4	12
198	10	2	3	10.98	4	12
199	10	2	3	12.43	4	12
200	10	2	3	11.25	4	12
201	10	2	3	12.36	4	12
202	19	2	3	14.27	4	12
203	19	2	3	14.36	4	12
204	19	2	3	11.64	4	12
205	19	2	3	9.52	4	12
206	17	2	3	15.25	4	12
207	17	2	3	13.49	4	12
208	17	2	3	11.79	4	12
209	17	2	3	10.26	4	12
210	16	2	3	14.89	4	12
211	16	2	3	10.41	4	12
212	16	2	3	12.88	4	12
213	16	2	3	13.16	4	12
214	16	2	3	11.19	4	12
215	16	2	3	9.17	4	12
216	16	2	3	10.27	4	12
217	16	2	3	11.47	4	12
218	13	2	3	9.67	5	12
219	13	2	3	12.97	5	12
220	13	2	3	16.3	5	12
221	13	2	3	10.84	5	12
222	13	2	3	13.51	5	12
223	10	2	3	10.79	4	12
224	10	2	3	12.06	4	12
225	10	2	3	12.49	4	12
226	10	2	3	10.46	4	12
227	15	2	3	12.23	4	12
228	15	2	3	13.65	4	12
229	15	2	3	12.98	4	12
230	15	2	3	10.49	4	12
231	20	2	3	11.78	4	12
232	20	2	3	16.33	4	12
233	20	2	3	10.02	4	12
234	20	2	3	13.81	4	12
235	6	2	1	9.01	4	4

## ANEXOS

236	6	2	1	10.53	4	4
237	6	2	1	11.01	4	4
238	6	2	1	17.11	4	4
239	10	2	1	11.41	4	4
240	10	2	1	11.6	4	4
241	10	2	1	9.94	4	4
242	10	2	1	10.83	4	4
243	7	2	1	10.69	5	4
244	7	2	1	12.89	5	4
245	7	2	1	10.95	5	4
246	7	2	1	11.41	5	4
247	7	2	1	9.64	5	4
248	18	2	1	13.41	5	4
249	18	2	1	11.79	5	4
250	18	2	1	12.13	5	4
251	18	2	1	12.92	5	4
252	18	2	1	14.99	5	4
253	13	2	1	10.44	3	4
254	13	2	1	11.58	3	4
255	13	2	1	12.22	3	4
256	12	2	1	10.77	5	4
257	12	2	1	11.11	5	4
258	12	2	1	10.9	5	4
259	12	2	1	10.23	5	4
260	12	2	1	9.66	5	4
261	8	2	1	8.63	5	4
262	8	2	1	10.27	5	4
263	8	2	1	9.69	5	4
264	8	2	1	10.28	5	4
265	8	2	1	13.34	5	4
266	12	2	1	10.91	5	4
267	12	2	1	11.48	5	4
268	12	2	1	10.56	5	4
269	12	2	1	9.69	5	4
270	12	2	1	10.52	5	4
271	11	2	2	10.5	4	5
272	11	2	2	13.25	4	5
273	11	2	2	11.56	4	5
274	11	2	2	12.65	4	5
275	15	2	2	12.34	4	5
276	15	2	2	11.47	4	5
277	15	2	2	17.79	4	5
278	15	2	2	14.1	4	5
279	14	2	2	14.42	4	5

## ANEXOS

280	14	2	2	12.88	4	5
281	14	2	2	11.82	4	5
282	14	2	2	11.37	4	5
283	12	2	2	11.64	4	5
284	12	2	2	12.07	4	5
285	12	2	2	13.64	4	5
286	12	2	2	13.71	4	5
287	11	2	2	14.45	3	5
288	11	2	2	12.49	3	5
289	11	2	2	14.52	3	5
290	12	2	2	13.97	3	5
291	12	2	2	13.88	3	5
292	12	2	2	12.17	3	5

### 7.1.3 Data de alumnos del curso “Análisis y Diseño Orientado a Objetos (AYD)”

#### Escuela Tecnológica sede Piura.

Alumno	Nota Requisito	Sede	Herramienta	Índice académico	Tamaño de grupo	Salón
293	15	3	3	5.65	4	10
294	15	3	3	10.72	4	10
295	15	3	3	11.04	4	10
296	15	3	3	11.06	4	10
297	15	3	3	11.05	5	10
298	15	3	3	8.8	5	10
299	15	3	3	9.96	5	10
300	15	3	3	10.32	5	10
301	15	3	3	8.77	5	10
302	14	3	3	10.85	4	10
303	14	3	3	10.05	4	10
304	14	3	3	10.34	4	10
305	14	3	3	6.42	4	10
306	14	3	3	8.08	4	10
307	14	3	3	12.51	4	10
308	14	3	3	11.1	4	10
309	14	3	3	10.43	4	10
310	14	3	3	7.95	4	10
311	14	3	3	10.09	4	10
312	14	3	3	6.59	4	10
313	14	3	3	7.9	4	10
314	16	3	3	10.49	4	10
315	16	3	3	10.62	4	10
316	16	3	3	11.98	4	10
317	16	3	3	5.67	4	10
318	13	3	3	9.2	4	10



## ANEXOS

319	13	3	3	9.49	4	10
320	13	3	3	12.48	4	10
321	13	3	3	8.53	4	10
322	13	3	3	9.85	3	10
323	13	3	3	9.62	3	10
324	13	3	3	8.26	3	10
325	13	3	3	13.09	4	9
326	13	3	3	13.21	4	9
327	13	3	3	12.03	4	9
328	13	3	3	11.38	4	9
329	8	3	3	11.71	4	9
330	8	3	3	11.48	4	9
331	8	3	3	10.83	4	9
332	8	3	3	8.54	4	9
333	14	3	3	11.12	4	9
334	14	3	3	10.76	4	9
335	14	3	3	12.07	4	9
336	14	3	3	13.09	4	9
337	8	3	1	11.60	4	1
338	8	3	1	7.95	4	1
339	8	3	1	10.90	4	1
340	8	3	1	10.09	4	1
341	9	3	1	9.39	4	1
342	9	3	1	11.76	4	1
343	9	3	1	10.99	4	1
344	9	3	1	9.55	4	1
345	13	3	1	8.54	4	1
346	13	3	1	10.89	4	1
347	13	3	1	11.28	4	1
348	13	3	1	11.32	4	1
349	11	3	1	10.55	4	1
350	11	3	1	13.05	4	1
351	11	3	1	10.62	4	1
352	11	3	1	9.44	4	1
353	10	3	1	12.53	5	1
354	10	3	1	11.52	5	1
355	10	3	1	11.25	5	1
356	10	3	1	12.42	5	1
357	10	3	1	11.22	5	1
358	13	3	2	13.23	3	6
359	13	3	2	11.72	3	6
360	13	3	2	11.49	3	6
361	14	3	2	12.48	3	6
362	14	3	2	13.44	3	6

## ANEXOS

363	14	3	2	13.26	3	6
364	13	3	2	12.28	4	6
365	13	3	2	12.48	4	6
366	13	3	2	9.65	4	6
367	13	3	2	12.36	4	6
368	12	3	2	10.28	4	6
369	12	3	2	13.55	4	6
370	12	3	2	8.95	4	6
371	12	3	2	9.85	4	6
372	13	3	2	10.22	4	6
373	13	3	2	12.55	4	6
374	13	3	2	12.85	4	6
375	13	3	2	12.5	4	6

## 7.2 Anexo 2: encuesta de métricas de requisitos (personal)

Las siguientes preguntas marcar 1 (ninguno), 2 (medio) y 3 (mucho)

### \*Obligatorio

#### 1. Tiene usted experiencia previa en Análisis de Software \*

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ Otro:

#### 2. Tiene usted experiencia previa en Diseño de Software \*

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ Otro:

#### 3. Tiene usted experiencia previa en Programación de Software \*

- ☒ 1
- ☐ 2
- ☐ 3
- ☐ Otro:

### Referidas al trabajo semestral – Especificación de Requisitos

#### 4. Qué herramienta usó para crear requisitos \*

- ☐ Microsoft Word
- ☐ Software REUSE Studio – Herramienta Case
- ☐ Software REUSE Studio con Patrones

#### 5. Cuánto tiempo lleva usted usando esta herramienta \*

- ☐ Menos de un año
- ☐ Menos de un mes
- ☐ Menos de una semana
- ☐ Más de un año

#### 6. Para su trabajo semestral, se apoyó en algún trabajo anterior \*

- ☐ Sí
- ☐ No

#### 7. Cuánto tiempo tardó en promedio crear un requisito con su herramienta \*

:  :

8. Para su trabajo semestral, cuánto tiempo tardó en promedio crear toda su lista de requisitos con su respectiva herramienta \*

 :  : 

Referidas a la gestión del requisito

9. Crear los requisitos le resultó \*

- ☐ Muy fácil
- ☐ Fácil
- ☐ Regular
- ☐ Difícil
- ☐ Muy difícil

10. La especificación de sus requisitos resultó ser \*

- ☐ Muy fácil
- ☐ Fácil
- ☐ Regular
- ☐ Difícil
- ☐ Muy difícil

11. Con cuántos requisitos cuenta su trabajo semestral \*

12. Tuvo usted que cambiar alguno de sus requisitos en algún momento \*

- ☐ Sí
- ☐ No

13. Cuánto tiempo tardo en modificar su lista de requisitos, si tuvo que hacerlo \*

 :  : 

14. Cuántas veces cambio su lista de requisitos \*

15. Si hubo cambios, cuántos de sus requisitos fueron modificados

16. Cuántos de sus requisitos fueron eliminados

17. Con la herramienta usada, qué tan fácil le resultó hacer dichos cambios \*

- ☐ Muy fácil

- ☐ ☐ Fácil
- ☐ ☐ Regular
- ☐ ☐ Difícil
- ☐ ☐ Muy difícil

### **Referidas a la Satisfacción del encuestado**

**18. Se sintió cómodo usando esta herramienta en la creación de sus requisitos \***

- ☐ ☐ Sí
- ☐ ☒ No
- ☐ ☐ Regular

**19. Cree usted que la herramienta que usó es la adecuada para el proceso de creación de requisitos \***

- ☐ ☐ Sí
- ☐ ☒ No

## ANEXOS

### 7.3 Anexo 3: detalle de las respuestas de cada alumno

Alumno	1. Tiene usted experiencia previa en Análisis de Software	2. Tiene usted experiencia previa en Diseño de Software	3. Tiene usted experiencia previa en Programación de Software	4. Qué herramienta usó para crear requisitos
1	2	1	2	Software REUSE Studio – Herramienta Case
2	2	2	1	Software REUSE Studio – Herramienta Case
3	3	3	2	Software REUSE Studio – Herramienta Case
4	2	2	2	Software REUSE Studio – Herramienta Case
5	2	2	1	Software REUSE Studio – Herramienta Case
6	2	2	2	Software REUSE Studio – Herramienta Case
7	2	2	2	Software REUSE Studio – Herramienta Case
8	2	2	3	Software REUSE Studio – Herramienta Case
9	2	2	1	Software REUSE Studio – Herramienta Case
10	2	2	1	Microsoft Word
11	2	2	1	Software REUSE Studio con Patrones
12	1	1	1	Software REUSE Studio con Patrones
13	2	1	1	Microsoft Word
14	3	2	2	Software REUSE Studio – Herramienta Case
15	2	2	1	Microsoft Word
16	2	1	1	Software REUSE Studio con Patrones
17	2	1	1	Microsoft Word
18	2	2	1	Software REUSE Studio – Herramienta Case
19	2	2	1	Software REUSE Studio con Patrones
20	2	2	1	Microsoft Word
21	2	2	1	Software REUSE Studio – Herramienta Case
22	2	2	2	Software REUSE Studio con Patrones
23	2	2	1	Microsoft Word
24	2	2	1	Software REUSE Studio – Herramienta Case

## ANEXOS

25	2	2	1	Software REUSE Studio – Herramienta Case
26	2	2	1	Microsoft Word
27	2	2	1	Software REUSE Studio – Herramienta Case
28	3	2	1	Software REUSE Studio con Patrones
29	2	2	2	Microsoft Word
30	2	2	1	Software REUSE Studio – Herramienta Case
31	2	2	1	Software REUSE Studio con Patrones
32	2	2	1	Microsoft Word
33	2	2	1	Software REUSE Studio – Herramienta Case
34	2	2	1	Microsoft Word
35	2	2	1	Software REUSE Studio – Herramienta Case
36	2	2	1	Software REUSE Studio con Patrones
37	2	2	2	Microsoft Word
38	2	2	1	Software REUSE Studio – Herramienta Case
39	2	2	1	Software REUSE Studio con Patrones
40	2	2	2	Microsoft Word
41	2	2	1	Software REUSE Studio – Herramienta Case
42	2	2	1	Software REUSE Studio con Patrones
43	2	2	2	Microsoft Word
44	2	2	1	Software REUSE Studio – Herramienta Case
45	2	2	1	Software REUSE Studio con Patrones
46	2	2	2	Microsoft Word
47	3	2	2	Software REUSE Studio – Herramienta Case
48	2	2	2	Software REUSE Studio con Patrones
49	2	2	1	Software REUSE Studio con Patrones
50	2	2	1	Microsoft Word
51	2	2	1	Software REUSE Studio con Patrones
52	2	2	1	Microsoft Word
53	2	2	1	Software REUSE Studio – Herramienta Case

## ANEXOS

54	2	2	1	Software REUSE Studio con Patrones
55	2	2	1	Microsoft Word
56	2	2	1	Software REUSE Studio – Herramienta Case
57	2	2	1	Software REUSE Studio con Patrones
58	2	2	1	Microsoft Word
59	2	2	1	Software REUSE Studio – Herramienta Case
60	2	2	1	Software REUSE Studio con Patrones
61	2	2	2	Microsoft Word
62	2	2	1	Software REUSE Studio – Herramienta Case
63	2	2	1	Software REUSE Studio con Patrones
64	2	2	1	Microsoft Word
65	2	2	2	Software REUSE Studio – Herramienta Case
66	2	2	1	Software REUSE Studio con Patrones
67	2	2	1	Software REUSE Studio con Patrones
68	2	2	1	Software REUSE Studio – Herramienta Case
69	2	2	1	Microsoft Word
70	2	2	1	Software REUSE Studio – Herramienta Case
71	2	2	1	Software REUSE Studio con Patrones
72	2	2	1	Microsoft Word
73	2	2	2	Microsoft Word
74	2	2	1	Software REUSE Studio – Herramienta Case
75	2	2	1	Software REUSE Studio – Herramienta Case
76	2	2	1	Microsoft Word
77	2	2	1	Software REUSE Studio con Patrones
78	2	2	1	Microsoft Word
79	2	2	1	Software REUSE Studio – Herramienta Case
80	2	2	2	Software REUSE Studio con Patrones
81	2	2	1	Software REUSE Studio con Patrones
82	3	3	2	Microsoft Word



## ANEXOS

83	2	2	1	Software REUSE Studio – Herramienta Case
84	2	2	1	Software REUSE Studio con Patrones
85	2	2	1	Microsoft Word
86	2	2	1	Microsoft Word
87	2	2	2	Software REUSE Studio – Herramienta Case
88	2	2	1	Software REUSE Studio con Patrones
89	2	2	1	Microsoft Word
90	2	2	1	Software REUSE Studio – Herramienta Case
91	2	2	1	Software REUSE Studio con Patrones
92	2	2	2	Microsoft Word
93	2	2	1	Software REUSE Studio – Herramienta Case
94	2	2	1	Software REUSE Studio con Patrones
95	2	2	1	Software REUSE Studio con Patrones
96	2	2	1	Microsoft Word
97	2	2	2	Software REUSE Studio – Herramienta Case
98	2	2	2	Software REUSE Studio con Patrones
99	2	2	1	Microsoft Word
100	2	2	1	Software REUSE Studio – Herramienta Case
101	2	2	1	Software REUSE Studio con Patrones
102	2	2	1	Microsoft Word
103	2	2	1	Software REUSE Studio – Herramienta Case
104	2	2	1	Software REUSE Studio – Herramienta Case
105	2	2	2	Software REUSE Studio – Herramienta Case
106	2	2	1	Microsoft Word
107	2	2	1	Software REUSE Studio – Herramienta Case
108	2	2	1	Software REUSE Studio con Patrones
109	2	2	2	Microsoft Word
110	2	2	2	Software REUSE Studio – Herramienta Case
111	2	2	2	Software REUSE Studio con

## ANEXOS

				Patrones
112	2	2	1	Microsoft Word
113	2	2	1	Software REUSE Studio – Herramienta Case
114	2	1	1	Software REUSE Studio con Patrones
115	2	1	1	Microsoft Word
116	2	1	1	Software REUSE Studio – Herramienta Case
117	2	1	1	Software REUSE Studio con Patrones
118	2	1	1	Microsoft Word
119	2	2	1	Software REUSE Studio – Herramienta Case
120	2	1	1	Software REUSE Studio con Patrones
121	2	2	2	Microsoft Word
122	2	2	1	Software REUSE Studio – Herramienta Case
123	2	1	1	Software REUSE Studio con Patrones
124	2	2	2	Microsoft Word
125		2	1	Software REUSE Studio – Herramienta Case
126	2	1	1	Software REUSE Studio con Patrones
127	2	2	2	Microsoft Word
128	2	2	1	Software REUSE Studio – Herramienta Case
129	2	2	1	Software REUSE Studio con Patrones
130	2	2	1	Microsoft Word
131	2	1	1	Software REUSE Studio – Herramienta Case
132	2	1	1	Software REUSE Studio con Patrones
133	2	2	1	Microsoft Word
134	2	2	1	Software REUSE Studio – Herramienta Case
135	2	1	1	Software REUSE Studio con Patrones
136	2	1	1	Microsoft Word
137	2	2	1	Software REUSE Studio – Herramienta Case
138	2	1	1	Software REUSE Studio con Patrones
139	2	1	1	Microsoft Word

## ANEXOS

140	2	1	1	Software REUSE Studio – Herramienta Case
141	2	1	1	Software REUSE Studio con Patrones
142	2	2	1	Microsoft Word
143	2	1	1	Software REUSE Studio – Herramienta Case
144	2	1	1	Software REUSE Studio con Patrones
145	2	1	1	Microsoft Word
146	2	1	1	Software REUSE Studio – Herramienta Case
147	2	1	1	Software REUSE Studio con Patrones
148	2	1	1	Microsoft Word
149	2	2	1	Software REUSE Studio – Herramienta Case
150	2	1	1	Software REUSE Studio con Patrones
151	2	1	1	Microsoft Word
152	2	2	1	Software REUSE Studio – Herramienta Case
153	2	1	1	Software REUSE Studio con Patrones
154	2	1	1	Microsoft Word
155	2	1	1	Software REUSE Studio – Herramienta Case
156	2	2	1	Software REUSE Studio con Patrones
157	2	1	1	Microsoft Word
158	2	2	2	Software REUSE Studio – Herramienta Case
159	2	1	1	Software REUSE Studio con Patrones
160	2	1	1	Microsoft Word
161	2	2	2	Software REUSE Studio – Herramienta Case
162	2	1	1	Software REUSE Studio con Patrones
163	2	1	1	Microsoft Word
164	2	1	1	Software REUSE Studio – Herramienta Case
165	2	2	1	Software REUSE Studio con Patrones
166	2	1	2	Microsoft Word
167	2	2	1	Software REUSE Studio – Herramienta Case
168	2	1	1	Software REUSE Studio con

## ANEXOS

				Patrones
169	2	2	2	Microsoft Word
170	2	1	1	Software REUSE Studio – Herramienta Case
171	2	2	1	Software REUSE Studio con Patrones
172	2	1	1	Microsoft Word
173	2	1	1	Software REUSE Studio – Herramienta Case
174	2	1	1	Software REUSE Studio con Patrones
175	2	1	1	Microsoft Word
176	2	2	2	Software REUSE Studio – Herramienta Case
177	2	1	1	Software REUSE Studio con Patrones
178	2	1	1	Microsoft Word
179	2	2	1	Software REUSE Studio – Herramienta Case
180	2	2	1	Software REUSE Studio con Patrones
181	2	2	1	Microsoft Word
182	2	1	1	Software REUSE Studio – Herramienta Case
183	2	1	1	Software REUSE Studio con Patrones
184	2	1	1	Microsoft Word
185	2	2	2	Software REUSE Studio – Herramienta Case
186	2	2	2	Software REUSE Studio con Patrones
187	2	1	1	Microsoft Word
188	2	2	2	Software REUSE Studio – Herramienta Case
189	2	2	1	Software REUSE Studio con Patrones
190	2	2	1	Microsoft Word
191	2	1	1	Software REUSE Studio – Herramienta Case
192	2	1	1	Software REUSE Studio con Patrones
193	2	1	1	Microsoft Word
194	2	1	1	Software REUSE Studio – Herramienta Case
195	2	1	1	Software REUSE Studio con Patrones
196	2	2	1	Microsoft Word

## ANEXOS

197	2	2	2	Software REUSE Studio – Herramienta Case
198	2	1	1	Software REUSE Studio con Patrones
199	2	1	1	Microsoft Word
200	2	2	2	Software REUSE Studio – Herramienta Case
201	2	1	1	Software REUSE Studio con Patrones
202	2	1	1	Microsoft Word
203	2	1	1	Software REUSE Studio – Herramienta Case
204	2	1	1	Software REUSE Studio con Patrones
205	2	2	2	Microsoft Word
206	2	1	1	Software REUSE Studio – Herramienta Case
207	2	1	1	Software REUSE Studio con Patrones
208	2	2	2	Microsoft Word
209	2	1	2	Software REUSE Studio – Herramienta Case
210	2	1	1	Software REUSE Studio con Patrones
211	2	1	1	Microsoft Word
212	2	1	1	Software REUSE Studio – Herramienta Case
213	2	1	1	Software REUSE Studio con Patrones
214	2	2	2	Microsoft Word
215	2	2	1	Software REUSE Studio – Herramienta Case
216	2	1	1	Software REUSE Studio con Patrones
217	2	2	2	Microsoft Word
218	2	1	1	Software REUSE Studio – Herramienta Case
219	2	1	1	Software REUSE Studio con Patrones
220	2	1	1	Software REUSE Studio con Patrones
221	2	2	1	Software REUSE Studio con Patrones
222	2	1	1	Software REUSE Studio con Patrones
223	2	2	2	Software REUSE Studio con Patrones
224	2	1	1	Software REUSE Studio con Patrones

## ANEXOS

225	2	1	1	Software REUSE Studio con Patrones
226	2	2	1	Software REUSE Studio con Patrones
227	2	1	1	Software REUSE Studio con Patrones
228	2	1	1	Software REUSE Studio con Patrones
229	2	1	1	Software REUSE Studio con Patrones
230	2	1	1	Software REUSE Studio con Patrones
231	2	1	1	Software REUSE Studio con Patrones
232	2	1	2	Software REUSE Studio con Patrones
233	2	1	1	Software REUSE Studio con Patrones
234	2	1	1	Software REUSE Studio con Patrones
235	2	2	1	Software REUSE Studio con Patrones
236	2	1	1	Software REUSE Studio con Patrones
237	2	2	1	Software REUSE Studio con Patrones
238	2	1	1	Software REUSE Studio con Patrones
239	2	1	1	Software REUSE Studio con Patrones
240	2	1	1	Software REUSE Studio con Patrones
241	2	2	1	Software REUSE Studio con Patrones
242	2	1	1	Software REUSE Studio con Patrones
243	2	1	2	Software REUSE Studio con Patrones
244	2	1	1	Software REUSE Studio con Patrones
245	2	1	1	Software REUSE Studio con Patrones
246	2	2	1	Software REUSE Studio con Patrones
247	2	2	1	Software REUSE Studio con Patrones
248	2	1	1	Software REUSE Studio con Patrones
249	2	1	1	Software REUSE Studio con Patrones
250	2	1	2	Software REUSE Studio con Patrones
251	2	2	1	Software REUSE Studio con

## ANEXOS

				Patrones
252	2	1	1	Software REUSE Studio con Patrones
253	2	2	1	Software REUSE Studio con Patrones
254	2	1	1	Software REUSE Studio con Patrones
255	2	1	1	Software REUSE Studio con Patrones
256	2	1	1	Software REUSE Studio con Patrones
257	2	1	1	Software REUSE Studio con Patrones
258	2	1	1	Software REUSE Studio con Patrones
259	2	1	1	Software REUSE Studio con Patrones
260	2	1	1	Software REUSE Studio con Patrones
261	2	1	1	Software REUSE Studio con Patrones
262	2	1	1	Software REUSE Studio con Patrones
263	2	1	1	Software REUSE Studio con Patrones
264	2	1	1	Software REUSE Studio con Patrones
265	2	1	1	Software REUSE Studio con Patrones
266	2	2	1	Software REUSE Studio con Patrones
267	2	2	1	Software REUSE Studio con Patrones
268	2	1	1	Software REUSE Studio con Patrones
269	3	2	1	Software REUSE Studio con Patrones
270	3	2	1	Software REUSE Studio con Patrones
271	3	2	1	Software REUSE Studio con Patrones
272	3	2	2	Software REUSE Studio con Patrones
273	3	2	2	Software REUSE Studio con Patrones
274	3	2	2	Software REUSE Studio con Patrones
275	3	2	2	Software REUSE Studio con Patrones
276	3	2	1	Software REUSE Studio con Patrones
277	2	1	1	Software REUSE Studio – Herramienta Case

## ANEXOS

278	2	2	1	Software REUSE Studio – Herramienta Case
279	2	2	2	Software REUSE Studio – Herramienta Case
280	2	2	1	Software REUSE Studio – Herramienta Case
281	2	2	1	Software REUSE Studio – Herramienta Case
282	2	2	1	Software REUSE Studio – Herramienta Case
283	2	2	3	Software REUSE Studio – Herramienta Case
284	2	2	2	Software REUSE Studio – Herramienta Case
285	2	2	1	Software REUSE Studio – Herramienta Case
286	3	2	1	Software REUSE Studio – Herramienta Case
287	3	2	1	Software REUSE Studio – Herramienta Case
288	2	2	1	Software REUSE Studio – Herramienta Case
289	2	2	1	Software REUSE Studio – Herramienta Case
290	3	2	1	Software REUSE Studio – Herramienta Case
291	2	2	1	Software REUSE Studio – Herramienta Case
292	3	2	1	Software REUSE Studio – Herramienta Case
293	2	2	1	Software REUSE Studio – Herramienta Case
294	2	2	1	Software REUSE Studio – Herramienta Case
295	2	2	1	Software REUSE Studio – Herramienta Case
296	2	2	1	Software REUSE Studio – Herramienta Case
297	3	2	1	Software REUSE Studio – Herramienta Case
298	2	2	1	Software REUSE Studio – Herramienta Case
299	2	1	1	Software REUSE Studio – Herramienta Case
300	2	2	1	Software REUSE Studio – Herramienta Case
301	2	1	1	Software REUSE Studio – Herramienta Case
302	2	1	1	Software REUSE Studio – Herramienta Case
303	2	2	1	Software REUSE Studio – Herramienta Case
304	2	2	1	Software REUSE Studio –



## ANEXOS

				Herramienta Case
305	2	2	1	Microsoft Word
306	2	2	1	Software REUSE Studio – Herramienta Case
307	2	2	1	Software REUSE Studio – Herramienta Case
308	2	2	1	Software REUSE Studio – Herramienta Case
309	3	2	1	Software REUSE Studio – Herramienta Case
310	2	1	1	Software REUSE Studio – Herramienta Case
311	2	1	1	Software REUSE Studio – Herramienta Case
312	2	2	1	Software REUSE Studio – Herramienta Case
313	2	2	1	Software REUSE Studio – Herramienta Case
314	2	1	1	Software REUSE Studio – Herramienta Case
315	2	2	1	Software REUSE Studio – Herramienta Case
316	2	2	2	Software REUSE Studio – Herramienta Case
317	3	2	1	Software REUSE Studio – Herramienta Case
318	2	2	1	Software REUSE Studio – Herramienta Case
319	3	2	1	Software REUSE Studio – Herramienta Case
320	2	2	1	Software REUSE Studio – Herramienta Case
321	3	1	1	Software REUSE Studio – Herramienta Case
322	2	3	3	Software REUSE Studio – Herramienta Case
323	2	2	1	Microsoft Word
324	2	2	1	Microsoft Word
325	2	1	1	Microsoft Word
326	2	2	1	Microsoft Word
327	3	2	2	Microsoft Word
328	2	1	1	Microsoft Word
329	3	2	2	Microsoft Word
330	3	3	2	Microsoft Word
331	2	1	1	Microsoft Word
332	3	3	2	Microsoft Word

## ANEXOS

333	2	2	2	Microsoft Word
334	3	2	1	Microsoft Word
335	2	3	3	Microsoft Word
336	2	1	1	Microsoft Word
337	2	2	1	Microsoft Word
338	2	2	1	Microsoft Word
339	3	2	2	Microsoft Word
340	3	3	3	Microsoft Word
341	2	2	1	Microsoft Word
342	2	2	1	Microsoft Word
343	3	3	2	Microsoft Word
344	3	3	1	Microsoft Word
345	3	3	2	Microsoft Word
346	2	2	2	Microsoft Word
347	3	2	1	Microsoft Word
348	3	2	1	Microsoft Word
349	2	3	2	Microsoft Word
350	2	2	2	Microsoft Word
351	3	3	1	Microsoft Word
352	2	2	1	Microsoft Word
353	2	3	3	Microsoft Word
354	2	1	1	Microsoft Word
355	2	2	1	Microsoft Word
356	2	2	1	Microsoft Word
357	2	2	1	Microsoft Word
358	3	2	2	Microsoft Word
359	2	2	1	Microsoft Word
360	3	2	2	Microsoft Word
361	2	2	1	Microsoft Word
362	2	2	2	Microsoft Word
363	2	3	1	Microsoft Word
364	3	3	1	Microsoft Word
365	2	3	1	Microsoft Word
366	2	2	1	Microsoft Word
367	2	2	1	Microsoft Word

## ANEXOS

368	2	3	1	Microsoft Word
369	3	3	1	Microsoft Word
370	2	2	2	Microsoft Word
371	2	3	1	Microsoft Word
372	3	3	1	Microsoft Word
373	3	2	1	Microsoft Word
374	3	2	1	Microsoft Word
375	2	3	1	Microsoft Word

Alumno	5. Cuánto tiempo lleva usted usando esta herramienta	6. Para su trabajo semestral, se apoyó en algún trabajo anterior	7. Cuánto tiempo tardó en promedio crear un requisito con su herramienta	8. Para su trabajo semestral, cuánto tiempo tardó en promedio crear toda su lista de requisitos con su respectiva herramienta
1	Menos de un año	No	10	1140
2	Menos de un mes	Sí	10	1440
3	Más de un año	Sí	50	330
4	Menos de un año	Sí	15	780
5	Menos de un mes	Sí	5	60
6	Menos de un año	Sí	5	420
7	Menos de un año	Sí	60	1080
8	Menos de un mes	Sí	5	185
9	Menos de un año	Sí	25	300
10	Más de un año	Sí	45	600
11	Menos de un mes	Sí	15	360
12	Menos de un mes	Sí	20	360
13	Más de un año	Sí	30	420
14	Menos de un año	Sí	10	480
15	Más de un año	Sí	40	780
16	Menos de un mes	Sí	25	480
17	Más de un año	Sí	35	780
18	Menos de un mes	Sí	30	540
19	Menos de un mes	Sí	15	180
20	Más de un año	Sí	35	780
21	Menos de un mes	No	20	420
22	Menos de un mes	No	5	360

## ANEXOS

23	Más de un año	Sí	35	960
24	Menos de un año	Sí	3	120
25	Menos de un año	Sí	10	300
26	Más de un año	Sí	43	750
27	Menos de un mes	Sí	20	360
28	Menos de un mes	Sí	10	720
29	Más de un año	Sí	40	840
30	Menos de un mes	Sí	15	420
31	Menos de un mes	Sí	10	240
32	Más de un año	Sí	30	480
33	Menos de un mes	Sí	23	540
34	Más de un año	Sí	38	420
35	Menos de un año	Sí	20	480
36	Menos de un mes	Sí	11	300
37	Más de un año	Sí	36	480
38	Menos de un año	Sí	30	540
39	Menos de un año	Sí	10	360
40	Más de un año	Sí	40	420
41	Menos de un año	Sí	20	420
42	Menos de un año	Sí	10	300
43	Más de un año	Sí	40	600
44	Menos de un año	Sí	25	420
45	Menos de un año	Sí	15	300
46	Más de un año	Sí	35	600
47	Menos de un año	Sí	25	360
48	Menos de un año	Sí	12	300
49	Menos de un año	No	10	480
50	Más de un año	Sí	45	660
51	Menos de un año	Sí	15	600
52	Más de un año	Sí	40	720
53	Menos de un año	Sí	25	420
54	Menos de un año	No	10	360
55	Más de un año	Sí	30	480
56	Menos de un año	Sí	20	420
57	Menos de un año	Sí	12	600

## ANEXOS

58	Más de un año	Sí	35	720
59	Menos de una semana	Sí	20	600
60	Menos de un año	Sí	15	180
61	Más de un año	Sí	20	360
62	Menos de un año	Sí	10	360
63	Menos de un año	Sí	30	720
64	Más de un año	Sí	20	360
65	Menos de un año	Sí	18	480
66	Menos de un año	Sí	8	360
67	Menos de un mes	Sí	5	180
68	Menos de un año	Sí	15	240
69	Más de un año	No	40	480
70	Menos de un año	Sí	22	600
71	Menos de un año	Sí	5	240
72	Más de un año	Sí	35	600
73	Más de un año	Sí	35	420
74	Menos de un año	Sí	23	420
75	Menos de un mes	Sí	5	180
76	Más de un año	Sí	10	180
77	Menos de un mes	Sí	3	120
78	Más de un año	Sí	90	600
79	Menos de un mes	Sí	180	1440
80	Menos de un mes	Sí	60	960
81	Menos de un mes	Sí	3	420
82	Más de un año	Sí	35	360
83	Menos de un mes	Sí	13	480
84	Menos de un año	Sí	50	1440
85	Más de un año	Sí	60	840
86	Más de un año	Sí	30	480
87	Menos de un mes	Sí	20	420
88	Menos de un mes	Sí	5	240
89	Más de un año	Sí	50	600
90	Menos de un año	Sí	25	720
91	Menos de un año	Sí	8	480

## ANEXOS

92	Más de un año	Sí	70	300
93	Menos de un año	Sí	40	600
94	Menos de un mes	Sí	3	300
95	Menos de un mes	Sí	7	300
96	Más de un año	Sí	120	720
97	Menos de un mes	Sí	17	600
98	Menos de un mes	Sí	6	240
99	Menos de un año	Sí	45	600
100	Menos de un mes	Sí	10	600
101	Menos de un año	No	7	480
102	Más de un año	Sí	60	780
103	Menos de un mes	Sí	3	360
104	Menos de un año	Sí	60	360
105	Menos de un mes	Sí	20	300
106	Menos de un mes	Sí	26	480
107	Menos de un mes	Sí	180	1440
108	Menos de un mes	Sí	30	360
109	Más de un año	Sí	30	660
110	Menos de un mes	Sí	23	420
111	Menos de un mes	Sí	15	600
112	Más de un año	Sí	45	720
113	Menos de un mes	Sí	20	480
114	Menos de un mes	Sí	12	300
115	Más de un año	Sí	60	900
116	Menos de un mes	Sí	25	480
117	Menos de un mes	Sí	4	360
118	Más de un año	Sí	70	780
119	Menos de un año	Sí	15	420
120	Menos de un mes	Sí	7	420
121	Más de un año	Sí	35	480
122	Menos de un mes	Sí	15	600
123	Menos de un mes	No	7	480
124	Más de un año	Sí	90	600
125	Menos de un mes	Sí	40	540
126	Menos de un mes	Sí	10	600

## ANEXOS

127	Más de un año	Sí	75	840
128	Menos de un mes	Sí	14	480
129	Menos de un mes	Sí	5	480
130	Más de un año	Sí	110	900
131	Menos de un mes	Sí	15	600
132	Menos de un mes	Sí	7	300
133	Menos de un año	Sí	60	1020
134	Menos de un mes	Sí	20	360
135	Menos de un mes	Sí	4	240
136	Más de un año	Sí	50	540
137	Menos de una semana	No	20	540
138	Menos de un mes	Sí	5	360
139	Más de un año	Sí	35	600
140	Menos de un mes	Sí	12	360
141	Menos de un mes	Sí	5	240
142	Más de un año	Sí	29	600
143	Menos de un año	Sí	15	480
144	Menos de un año	Sí	7	360
145	Más de un año	Sí	32	720
146	Menos de un mes	Sí	20	480
147	Menos de un mes	Sí	4	240
148	Más de un año	Sí	50	720
149	Menos de un mes	Sí	20	360
150	Menos de un mes	Sí	7	240
151	Más de un año	Sí	60	900
152	Menos de un mes	Sí	25	600
153	Menos de un mes	Sí	6	240
154	Más de un año	Sí	60	900
155	Menos de un mes	Sí	14	360
156	Menos de un mes	Sí	5	540
157	Más de un año	Sí	50	1200
158	Menos de un mes	Sí	15	420
159	Menos de un mes	Sí	10	1080
160	Más de un año	Sí	120	1440

## ANEXOS

161	Menos de un mes	Sí	10	480
162	Menos de un año	Sí	5	360
163	Más de un año	Sí	60	780
164	Menos de un mes	Sí	35	600
165	Menos de un mes	Sí	12	420
166	Más de un año	Sí	60	900
167	Menos de un mes	Sí	35	600
168	Menos de un mes	Sí	10	480
169	Más de un año	Sí	70	1020
170	Menos de un mes	Sí	30	600
171	Menos de un mes	Sí	6	420
172	Más de un año	Sí	60	1020
173	Menos de un mes	Sí	11	600
174	Menos de un mes	Sí	5	480
175	Más de un año	Sí	50	840
176	Menos de un mes	Sí	29	480
177	Menos de un año	Sí	4	300
178	Más de un año	Sí	70	1200
179	Menos de un mes	Sí	15	360
180	Menos de un mes	Sí	3	300
181	Más de un año	Sí	90	1380
182	Menos de un mes	Sí	10	300
183	Más de un año	Sí	8	420
184	Más de un año	Sí	60	720
185	Menos de un mes	Sí	10	360
186	Menos de un mes	Sí	10	720
187	Más de un año	Sí	30	900
188	Menos de un mes	Sí	30	900
189	Menos de un mes	Sí	10	360
190	Más de un año	Sí	45	1080
191	Menos de un mes	Sí	15	420
192	Menos de un mes	Sí	7	240
193	Más de un año	Sí	40	840
194	Menos de un mes	Sí	13	540
195	Menos de un mes	Sí	10	360



## ANEXOS

196	Más de un año	Sí	60	840
197	Menos de un mes	Sí	9	600
198	Menos de un mes	Sí	6	300
199	Más de un año	Sí	60	600
200	Menos de un mes	Sí	15	420
201	Menos de un mes	Sí	5	240
202	Más de un año	Sí	60	960
203	Menos de un mes	Sí	15	600
204	Menos de un mes	Sí	4	360
205	Más de un año	Sí	40	840
206	Menos de un mes	Sí	20	420
207	Menos de un mes	Sí	10	240
208	Más de un año	Sí	42	540
209	Menos de un mes	Sí	19	420
210	Menos de un mes	Sí	10	240
211	Más de un año	Sí	30	540
212	Menos de un mes	Sí	20	408
213	Menos de un mes	Sí	12	180
214	Más de un año	Sí	40	300
215	Menos de un mes	Sí	20	480
216	Menos de un mes	Sí	10	240
217	Más de un año	Sí	40	600
218	Menos de un mes	Sí	21	480
219	Menos de un mes	Sí	15	360
220	Menos de un mes	Sí	10	180
221	Menos de un mes	Sí	10	240
222	Menos de un mes	Sí	15	420
223	Menos de un año	Sí	14	300
224	Menos de un mes	Sí	10	420
225	Menos de un año	Sí	20	600
226	Menos de un mes	Sí	15	540
227	Menos de un mes	Sí	10	300
228	Menos de un año	Sí	5	240
229	Más de un año	Sí	30	720
230	Menos de un mes	Sí	80	600

## ANEXOS

231	Menos de un año	No	18	480
232	Menos de un mes	Sí	8	240
233	Menos de un mes	Sí	15	360
234	Menos de un mes	Sí	9	420
235	Menos de un mes	Sí	10	490
236	Menos de un mes	Sí	7	180
237	Menos de un mes	Sí	15	420
238	Menos de un mes	Sí	5	180
239	Menos de un mes	Sí	25	300
240	Menos de una semana	Sí	10	540
241	Menos de un mes	Sí	9	240
242	Menos de una semana	Sí	6	240
243	Menos de un mes	Sí	10	300
244	Menos de un mes	Sí	15	360
245	Menos de un mes	Sí	5	240
246	Menos de un mes	Sí	11	300
247	Menos de un mes	Sí	5	240
248	Menos de un mes	Sí	9	600
249	Menos de un mes	Sí	20	360
250	Menos de un año	Sí	13	420
251	Más de un año	Sí	8	360
252	Menos de un mes	Sí	9	660
253	Menos de un año	Sí	15	600
254	Menos de un año	Sí	10	240
255	Menos de un mes	Sí	20	360
256	Menos de un mes	Sí	15	300
257	Menos de un mes	Sí	7	360
258	Menos de una semana	Sí	20	780
259	Menos de un mes	Sí	17	540
260	Menos de un mes	Sí	11	540
261	Menos de un mes	Sí	15	360
262	Menos de un mes	Sí	15	300
263	Menos de un mes	No	10	480
264	Menos de un mes	Sí	7	420

## ANEXOS

265	Menos de un mes	Sí	10	240
266	Menos de un mes	Sí	5	240
267	Menos de un mes	Sí	7	240
268	Menos de un mes	Sí	15	300
269	Menos de un mes	Sí	10	480
270	Menos de un mes	Sí	10	240
271	Menos de un mes	Sí	25	960
272	Menos de un mes	Sí	20	720
273	Menos de un mes	Sí	5	240
274	Menos de un mes	Sí	10	420
275	Menos de un mes	Sí	10	480
276	Menos de un mes	Sí	60	600
277	Menos de una semana	Sí	20	480
278	Menos de un mes	Sí	26	960
279	Menos de un mes	Sí	35	900
280	Menos de un mes	Sí	25	540
281	Menos de un mes	Sí	30	780
282	Menos de un mes	Sí	20	720
283	Menos de un mes	Sí	25	240
284	Menos de un mes	Sí	20	780
285	Menos de un mes	Sí	30	900
286	Menos de un mes	Sí	25	600
287	Menos de un mes	Sí	10	840
288	Menos de un mes	Sí	15	720
289	Menos de un mes	Sí	15	600
290	Menos de un mes	Sí	25	900
291	Menos de un mes	Sí	20	960
292	Menos de un año	Sí	30	1020
293	Menos de un mes	Sí	20	720
294	Menos de un mes	Sí	15	600
295	Menos de un mes	Sí	35	720
296	Menos de un mes	Sí	20	900
297	Menos de un mes	Sí	10	300
298	Menos de un mes	Sí	40	720

## ANEXOS

299	Menos de un mes	Sí	21	900
300	Menos de un mes	Sí	20	420
301	Menos de un mes	Sí	20	900
302	Menos de un mes	Sí	20	600
303	Menos de un mes	Sí	30	300
304	Menos de un mes	Sí	15	840
305	Menos de un año	Sí	5	720
306	Menos de un mes	Sí	20	4200
307	Menos de un año	No	5	1440
308	Menos de un mes	Sí	25	660
309	Menos de un mes	Sí	20	360
310	Menos de un mes	Sí	25	480
311	Menos de un mes	Sí	30	600
312	Menos de un año	No	12	480
313	Menos de un mes	Sí	40	960
314	Menos de un mes	Sí	15	240
315	Menos de un mes	Sí	23	720
316	Menos de un mes	Sí	25	900
317	Menos de un mes	Sí	25	720
318	Menos de una semana	Sí	20	300
319	Menos de un mes	Sí	60	780
320	Menos de una semana	Sí	20	540
321	Menos de un mes	Sí	30	300
322	Menos de un mes	Sí	10	300
323	Menos de un mes	Sí	25	960
324	Menos de un mes	Sí	20	470
325	Más de un año	Sí	60	600
326	Más de un año	Sí	50	480
327	Más de un año	Sí	50	840
328	Más de un año	Sí	40	750
329	Más de un año	Sí	45	900
330	Más de un año	No	55	480
331	Más de un año	Sí	30	660
332	Más de un año	Sí	55	600

## ANEXOS

333	Más de un año	Sí	60	540
334	Más de un año	Sí	50	600
335	Más de un año	Sí	120	660
336	Más de un año	Sí	40	420
337	Más de un año	Sí	70	480
338	Menos de un mes	Sí	65	480
339	Más de un año	Sí	30	540
340	Más de un año	Sí	55	660
341	Más de un año	Sí	30	600
342	Más de un año	Sí	40	840
343	Más de un año	Sí	55	420
344	Más de un año	Sí	90	540
345	Más de un año	Sí	100	420
346	Más de un año	Sí	75	360
347	Más de un año	Sí	60	7520
348	Más de un año	Sí	40	300
349	Más de un año	Sí	50	360
350	Más de un año	Sí	30	360
351	Más de un año	Sí	60	720
352	Más de un año	Sí	40	420
353	Más de un año	Sí	70	360
354	Más de un año	Sí	50	900
355	Más de un año	Sí	60	630
356	Más de un año	Sí	55	960
357	Más de un año	Sí	30	420
358	Más de un año	Sí	45	480
359	Más de un año	Sí	70	720
360	Más de un año	Sí	60	780
361	Más de un año	No	70	840
362	Más de un año	Sí	54	420
363	Más de un año	Sí	40	540
364	Más de un año	Sí	53	720
365	Más de un año	Sí	90	780
366	Más de un año	Sí	30	480
367	Más de un año	Sí	90	480

## ANEXOS

368	Más de un año	No	50	480
369	Más de un año	Sí	50	420
370	Más de un año	Sí	45	360
371	Más de un año	Sí	40	600
372	Más de un año	Sí	35	420
373	Más de un año	Sí	40	360
374	Más de un año	Sí	50	420
375	Más de un año	Sí	40	600

Alumno	9. Crear los requisitos le resultó	10. La especificación de sus requisitos resultó ser	11. Con cuántos requisitos cuenta su trabajo semestral	12. Tuvo usted que cambiar alguno de sus requisitos en algún momento	13. Cuánto tiempo tardó en modificar su lista de requisitos, si tuvo que hacerlo
1	Regular	Difícil	36	Sí	960
2	Regular	Difícil	20	Sí	180
3	Regular	Fácil	39	Sí	120
4	Regular	Regular	12	Sí	180
5	Fácil	Difícil	23	Sí	270
6	Regular	Regular	24	Sí	110
7	Regular	Difícil	8	Sí	720
8	Regular	Regular	27	Sí	735
9	Regular	Regular	35	Sí	300
10	Difícil	Difícil	31	Sí	420
11	Fácil	Fácil	67	Sí	180
12	Fácil	Fácil	40	Sí	120
13	Muy difícil	Difícil	47	Sí	240
14	Muy fácil	Muy fácil	64	Sí	120
15	Difícil	Difícil	49	Sí	300
16	Fácil	Regular	55	Sí	120
17	Difícil	Muy difícil	33	Sí	300
18	Fácil	Regular	65	Sí	180
19	Muy fácil	Muy fácil	73	Sí	60
20	Muy difícil	Muy difícil	39	Sí	600
21	Regular	Fácil	55	Sí	180

## ANEXOS

22	Muy fácil	Muy fácil	67	No	720
23	Muy difícil	Muy difícil	49	Sí	480
24	Regular	Regular	-	Sí	120
25	Regular	Regular	20	Sí	60
26	Difícil	Muy difícil	50	Sí	390
27	Fácil	Regular	68	No	720
28	Muy fácil	Fácil	70	Sí	180
29	Regular	Regular	60	Sí	360
30	Fácil	Regular	67	Sí	300
31	Muy fácil	Fácil	70	Sí	60
32	Muy difícil	Difícil	46	Sí	760
33	Fácil	Muy fácil	60	Sí	75
34	Difícil	Difícil	50	Sí	759
35	Fácil	Regular	75	Sí	760
36	Muy fácil	Fácil	80	Sí	745
37	Difícil	Regular	46	Sí	770
38	Regular	Regular	68	Sí	770
39	Fácil	Muy fácil	77	Sí	750
40	Regular	Regular	55	Sí	115
41	Fácil	Fácil	65	Sí	765
42	Muy fácil	Muy fácil	85	Sí	755
43	Muy difícil	Muy difícil	50	Sí	120
44	Fácil	Regular	75	Sí	770
45	Muy fácil	Muy fácil	76	Sí	60
46	Muy difícil	Difícil	40	Sí	80
47	Fácil	Regular	65	Sí	60
48	Muy fácil	Regular	85	Sí	60
49	Muy fácil	Fácil	85	Sí	120
50	Regular	Difícil	56	Sí	180
51	Regular	Regular	45	Sí	120
52	Regular	Difícil	50	Sí	120
53	Fácil	Regular	68	Sí	60
54	Muy fácil	Muy fácil	85	Sí	80
55	Difícil	Difícil	50	Sí	170
56	Regular	Regular	65	Sí	60

## ANEXOS

57	Muy fácil	Difícil	90	Sí	120
58	Regular	Regular	65	Sí	180
59	Muy fácil	Regular	70	Sí	100
60	Muy fácil	Muy fácil	60	Sí	760
61	Muy fácil	Muy fácil	60	Sí	60
62	Muy fácil	Muy fácil	70	Sí	90
63	Regular	Regular	68	Sí	240
64	Fácil	Fácil	63	Sí	120
65	Fácil	Fácil	76	Sí	180
66	Muy fácil	Muy fácil	75	Sí	60
67	Muy fácil	Muy fácil	73	No	720
68	Fácil	Regular	69	No	720
69	Difícil	Muy difícil	37	Sí	120
70	Fácil	Fácil	65	No	720
71	Muy fácil	Muy fácil	60	Sí	750
72	Regular	Difícil	50	Sí	180
73	Regular	Difícil	48	Sí	120
74	Fácil	Fácil	70	Sí	90
75	Fácil	Fácil	36	Sí	60
76	Regular	Regular	39	Sí	120
77	Muy fácil	Muy fácil	54	Sí	60
78	Difícil	Muy difícil	60	Sí	600
79	Regular	Difícil	50	Sí	600
80	Fácil	Regular	80	Sí	300
81	Muy fácil	Fácil	75	Sí	180
82	Regular	Regular	42	Sí	120
83	Fácil	Fácil	77	Sí	210
84	Fácil	Fácil	88	No	720
85	Regular	Regular	79	Sí	420
86	Difícil	Difícil	37	Sí	180
87	Regular	Fácil	53	Sí	240
88	Muy fácil	Fácil	86	Sí	120
89	Difícil	Difícil	32	Sí	180
90	Fácil	Fácil	55	No	720
91	Fácil	Muy fácil	73	Sí	240



## ANEXOS

92	Fácil	Muy fácil	61	Si	180
93	Fácil	Regular	55	Si	240
94	Muy fácil	Muy fácil	79	No	720
95	Muy fácil	Regular	72	Si	90
96	Muy difícil	Muy difícil	41	Si	300
97	Regular	Fácil	68	No	720
98	Muy fácil	Muy fácil	94	Si	180
99	Regular	Difícil	48	Si	300
100	Fácil	Regular	62	Si	240
101	Fácil	Fácil	82	Si	120
102	Difícil	Difícil	42	Si	240
103	Fácil	Fácil	73	Si	180
104	Regular	Regular	12	Si	750
105	Regular	Regular	25	Si	600
106	Difícil	Muy difícil	70	Si	360
107	Regular	Regular	70	Si	600
108	Regular	Regular	55	Si	600
109	Muy difícil	Muy difícil	37	Si	360
110	Regular	Regular	69	Si	180
111	Fácil	Muy difícil	75	Si	180
112	Difícil	Difícil	54	Si	300
113	Fácil	Regular	72	Si	240
114	Muy fácil	Fácil	82	Si	210
115	Difícil	Muy difícil	39	Si	420
116	Fácil	Fácil	63	Si	300
117	Muy fácil	Muy fácil	74	Si	100
118	Regular	Difícil	45	Si	240
119	Fácil	Muy difícil	62	Si	180
120	Muy fácil	Fácil	77	Si	120
121	Regular	Regular	39	Si	120
122	Fácil	Fácil	66	Si	300
123	Muy fácil	Muy fácil	84	Si	210
124	Regular	Difícil	32	Si	240
125	Fácil	Regular	67	Si	180
126	Fácil	Muy fácil	80	Si	240

## ANEXOS

127	Difícil	Difícil	47	Sí	600
128	Fácil	Regular	54	Sí	180
129	Muy fácil	Muy fácil	79	Sí	150
130	Difícil	Difícil	40	Sí	300
131	Fácil	Regular	63	Sí	210
132	Muy fácil	Muy fácil	72	Sí	120
133	Difícil	Muy difícil	43	Sí	300
134	Fácil	Fácil	70	Sí	120
135	Muy fácil	Muy fácil	76	Sí	120
136	Difícil	Muy difícil	39	Sí	300
137	Fácil	Fácil	60	Sí	680
138	Muy fácil	Muy fácil	80	Sí	60
139	Regular	Difícil	49	Sí	120
140	Fácil	Fácil	58	Sí	180
141	Muy fácil	Muy fácil	74	Sí	100
142	Regular	Regular	59	Sí	360
143	Fácil	Regular	62	Sí	150
144	Muy fácil	Muy fácil	79	Sí	120
145	Difícil	Muy difícil	49	Sí	300
146	Fácil	Fácil	56	Sí	150
147	Muy fácil	Muy fácil	64	Sí	60
148	Muy difícil	Muy difícil	40	Sí	600
149	Fácil	Regular	61	Sí	120
150	Regular	Regular	63	Sí	180
151	Regular	Difícil	58	Sí	240
152	Fácil	Regular	63	Sí	210
153	Muy fácil	Muy fácil	72	Sí	100
154	Regular	Muy difícil	75	Sí	360
155	Regular	Difícil	47	Sí	150
156	Muy fácil	Muy fácil	83	Sí	240
157	Muy difícil	Difícil	54	Sí	540
158	Regular	Regular	81	Sí	180
159	Difícil	Difícil	51	Sí	300
160	Difícil	Muy difícil	70	Sí	840
161	Fácil	Muy difícil	76	Sí	240

## ANEXOS

162	Muy fácil	Muy fácil	82	Si	120
163	Regular	Muy difícil	58	Si	240
164	Regular	Regular	66	Si	110
165	Muy fácil	Muy fácil	69	Si	180
166	Regular	Difícil	53	Si	120
167	Regular	Regular	67	Si	120
168	Fácil	Muy fácil	73	Si	180
169	Muy difícil	Muy difícil	54	Si	300
170	Fácil	Regular	68	Si	60
171	Muy fácil	Muy fácil	78	Si	180
172	Regular	Regular	40	Si	180
173	Fácil	Fácil	76	Si	240
174	Fácil	Muy fácil	76	Si	120
175	Regular	Regular	34	Si	240
176	Fácil	Fácil	55	Si	180
177	Muy fácil	Muy fácil	78	Si	120
178	Regular	Difícil	43	Si	300
179	Regular	Fácil	70	Si	180
180	Muy fácil	Muy fácil	86	Si	60
181	Difícil	Muy difícil	42	Si	240
182	Fácil	Muy difícil	60	Si	180
183	Muy fácil	Muy fácil	73	Si	120
184	Difícil	Muy difícil	45	Si	300
185	Fácil	Regular	58	Si	120
186	Regular	Regular	60	No	720
187	Difícil	Muy difícil	46	Si	300
188	Regular	Regular	60	Si	120
189	Muy fácil	Muy fácil	77	Si	120
190	Difícil	Muy difícil	48	Si	300
191	Fácil	Difícil	55	Si	120
192	Muy fácil	Muy fácil	75	Si	120
193	Regular	Regular	42	Si	300
194	Regular	Regular	60	Si	180
195	Fácil	Difícil	79	Si	300
196	Regular	Difícil	59	Si	360

## ANEXOS

197	Fácil	Muy fácil	46	Sí	240
198	Muy fácil	Muy fácil	75	Sí	120
199	Regular	Regular	47	Sí	420
200	Fácil	Fácil	70	Sí	360
201	Muy fácil	Muy fácil	80	Sí	180
202	Difícil	Difícil	53	Sí	360
203	Fácil	Regular	57	Sí	120
204	Muy fácil	Muy fácil	72	Sí	180
205	Difícil	Difícil	46	Sí	420
206	Fácil	Regular	72	Sí	60
207	Muy fácil	Muy fácil	83	Sí	750
208	Regular	Muy difícil	40	Sí	240
209	Fácil	Regular	65	Sí	60
210	Muy fácil	Muy fácil	72	Sí	120
211	Difícil	Muy difícil	42	Sí	240
212	Difícil	Difícil	64	Sí	300
213	Muy fácil	Difícil	78	Sí	180
214	Difícil	Muy difícil	38	Sí	180
215	Fácil	Regular	65	Sí	120
216	Muy fácil	Muy fácil	75	Sí	60
217	Muy difícil	Muy difícil	42	Sí	300
218	Regular	Difícil	69	Sí	60
219	Regular	Regular	50	Sí	180
220	Muy fácil	Muy fácil	77	Sí	120
221	Muy fácil	Difícil	74	Sí	120
222	Fácil	Muy difícil	84	Sí	240
223	Muy fácil	Fácil	71	Sí	240
224	Muy fácil	Fácil	69	Sí	120
225	Fácil	Muy difícil	78	Sí	240
226	Muy fácil	Muy fácil	83	Sí	300
227	Muy fácil	Fácil	67	Sí	180
228	Muy fácil	Difícil	80	Sí	150
229	Muy fácil	Regular	54	Sí	360
230	Muy fácil	Fácil	60	Sí	300
231	Fácil	Fácil	76	Sí	420

## ANEXOS

232	Muy fácil	Muy fácil	74	Si	130
233	Muy fácil	Fácil	80	Si	180
234	Muy fácil	Muy fácil	78	Si	180
235	Fácil	Fácil	73	Si	360
236	Muy fácil	Muy fácil	77	Si	60
237	Muy fácil	Muy difícil	84	Si	240
238	Muy fácil	Muy fácil	72	Si	75
239	Muy fácil	Muy fácil	76	Si	90
240	Fácil	Fácil	56	Si	120
241	Muy fácil	Fácil	79	Si	120
242	Muy fácil	Muy fácil	63	Si	180
243	Fácil	Muy fácil	65	Si	120
244	Fácil	Fácil	63	Si	120
245	Muy fácil	Difícil	64	Si	120
246	Muy fácil	Muy fácil	56	Si	110
247	Muy fácil	Difícil	85	Si	160
248	Fácil	Fácil	66	Si	240
249	Muy fácil	Fácil	68	Si	180
250	Muy fácil	Fácil	73	Si	220
251	Muy fácil	Fácil	83	Si	300
252	Fácil	Regular	61	Si	180
253	Muy fácil	Fácil	70	Si	240
254	Muy fácil	Muy fácil	68	Si	150
255	Fácil	Regular	50	Si	240
256	Muy fácil	Muy fácil	65	Si	240
257	Muy fácil	Muy fácil	59	Si	90
258	Fácil	Regular	65	Si	240
259	Muy fácil	Muy fácil	43	Si	180
260	Fácil	Difícil	83	Si	130
261	Muy fácil	Muy fácil	82	Si	160
262	Muy fácil	Fácil	74	Si	75
263	Fácil	Muy difícil	70	Si	180
264	Muy fácil	Muy difícil	85	Si	240
265	Muy fácil	Regular	55	Si	60
266	Muy fácil	Regular	68	Si	90

## ANEXOS

267	Muy fácil	Muy fácil	53	Sí	110
268	Muy fácil	Fácil	63	Sí	120
269	Muy fácil	Muy fácil	65	Sí	180
270	Fácil	Fácil	76	Sí	240
271	Regular	Regular	60	Sí	300
272	Fácil	Fácil	62	Sí	180
273	Muy fácil	Muy fácil	85	Sí	150
274	Muy fácil	Muy fácil	68	Sí	180
275	Muy fácil	Fácil	75	Sí	300
276	Fácil	Fácil	68	Sí	240
277	Regular	Regular	47	Sí	240
278	Regular	Regular	50	Sí	300
279	Fácil	Regular	48	Sí	770
280	Fácil	Fácil	47	Sí	240
281	Regular	Regular	45	Sí	180
282	Fácil	Regular	48	Sí	300
283	Fácil	Muy difícil	50	Sí	240
284	Fácil	Regular	55	Sí	60
285	Fácil	Regular	48	Sí	300
286	Fácil	Muy difícil	47	Sí	180
287	Muy fácil	Regular	48	Sí	300
288	Fácil	Regular	53	Sí	310
289	Fácil	Fácil	48	Sí	180
290	Regular	Difícil	38	Sí	360
291	Regular	Regular	49	Sí	300
292	Difícil	Difícil	52	Sí	240
293	Fácil	Fácil	63	Sí	180
294	Muy fácil	Muy fácil	73	Sí	170
295	Fácil	Fácil	57	Sí	270
296	Fácil	Fácil	57	Sí	370
297	Muy fácil	Muy fácil	60	Sí	180
298	Regular	Difícil	64	Sí	260
299	Fácil	Regular	59	Sí	600
300	Fácil	Fácil	47	Sí	240
301	Fácil	Fácil	46	Sí	180

## ANEXOS

302	Regular	Regular	67	Si	160
303	Fácil	Regular	51	Si	300
304	Regular	Difícil	40	Si	210
305	Regular	Difícil	40	Si	740
306	Fácil	Fácil	37	Si	220
307	Regular	Regular	30	Si	120
308	Regular	Regular	48	Si	180
309	Fácil	Regular	63	Si	60
310	Regular	Regular	49	Si	180
311	Fácil	Fácil	47	Si	210
312	Muy fácil	Muy fácil	57	Si	110
313	Difícil	Muy difícil	46	Si	360
314	Fácil	Fácil	36	Si	150
315	Fácil	Regular	51	Si	210
316	Fácil	Fácil	47	Si	240
317	Fácil	Regular	52	Si	195
318	Fácil	Regular	50	Si	310
319	Muy fácil	Muy fácil	65	Si	360
320	Fácil	Regular	38	Si	180
321	Fácil	Muy difícil	40	Si	180
322	Muy fácil	Muy fácil	46	Si	180
323	Fácil	Regular	55	Si	300
324	Fácil	Regular	68	Si	330
325	Regular	Difícil	47	Si	120
326	Difícil	Difícil	43	Si	420
327	Difícil	Muy difícil	40	Si	240
328	Regular	Difícil	42	Si	440
329	Difícil	Muy difícil	54	Si	240
330	Muy difícil	Muy difícil	53	Si	230
331	Difícil	Difícil	47	Si	360
332	Difícil	Muy difícil	40	Si	180
333	Fácil	Fácil	48	Si	240
334	Regular	Regular	49	Si	210
335	Muy difícil	Muy difícil	50	Si	240
336	Difícil	Difícil	47	Si	120

## ANEXOS

337	Muy difícil	Muy difícil	51	Sí	180
338	Difícil	Muy difícil	50	Sí	240
339	Regular	Difícil	55	Sí	240
340	Regular	Regular	60	Sí	180
341	Regular	Muy difícil	57	Sí	180
342	Difícil	Difícil	48	Sí	220
343	Muy difícil	Muy difícil	53	Sí	190
344	Difícil	Difícil	40	Sí	150
345	Difícil	Muy difícil	56	Sí	235
346	Difícil	Difícil	39	Sí	100
347	Muy difícil	Muy difícil	48	Sí	300
348	Difícil	Difícil	45	Sí	120
349	Difícil	Difícil	48	Sí	240
350	Regular	Muy difícil	63	Sí	180
351	Muy difícil	Muy difícil	51	Sí	226
352	Difícil	Muy difícil	60	Sí	190
353	Difícil	Difícil	45	Sí	360
354	Difícil	Muy difícil	58	Sí	210
355	Muy difícil	Muy difícil	49	Sí	240
356	Muy difícil	Muy difícil	62	Sí	300
357	Regular	Difícil	60	Sí	170
358	Difícil	Difícil	57	No	180
359	Muy difícil	Muy difícil	50	Sí	170
360	Muy difícil	Muy difícil	60	Sí	360
361	Muy difícil	Muy difícil	65	Sí	240
362	Difícil	Difícil	41	Sí	150
363	Regular	Difícil	52	Sí	220
364	Difícil	Difícil	58	Sí	185
365	Difícil	Muy difícil	60	Sí	300
366	Difícil	Muy difícil	42	Sí	60
367	Muy difícil	Muy difícil	40	Sí	180
368	Difícil	Difícil	51	Sí	300
369	Muy difícil	Muy difícil	48	Sí	180
370	Regular	Regular	37	Sí	240
371	Difícil	Difícil	41	Sí	160



## ANEXOS

372	Regular	Difícil	37	Sí	190
373	Difícil	Difícil	36	Sí	120
374	Regular	Difícil	40	Sí	150
375	Difícil	Muy difícil	38	Sí	180

Alumno	14. Cuántas veces cambio su lista de requisitos	15. Si hubo cambios, cuántos de sus requisitos fueron modificados	16. Cuántos de sus requisitos fueron eliminados	17. Con la herramienta usada, qué tan fácil le resultó hacer dichos cambios
1	2	5	3	Regular
2	2	7	5	Regular
3	2	12	0	Fácil
4	4	4	5	Regular
5	2	0	0	Regular
6	2	6	0	Regular
7	1	3	2	Fácil
8	1	4	7	Regular
9	3	20	7	Regular
10	4	27	10	Difícil
11	3	37	8	Muy fácil
12	2	15	6	Fácil
13	3	23	17	Muy fácil
14	2	30	5	Fácil
15	4	28	12	Difícil
16	2	10	7	Fácil
17	2	15	3	Muy difícil
18	2	12	7	Fácil
19	2	25	14	Muy fácil
20	5	20	15	Muy difícil
21	3	20	4	Fácil
22	0	0	5	Muy fácil
23	4	40	10	Muy difícil
24	-	-	-	Fácil
25	3	10	3	Muy fácil
26	2	35	12	Difícil

## ANEXOS

27	0			Regular
28	2	26	11	Muy fácil
29	3	17	5	Regular
30	3	25	10	Fácil
31	2	10	5	Muy fácil
32	3	8	3	Regular
33	2	12	5	Fácil
34	2	7	4	Difícil
35	2	8	4	Muy fácil
36	2	5	4	Muy fácil
37	3	9	7	Difícil
38	2	15	8	Regular
39	2	6	5	Muy fácil
40	2	10	6	Difícil
41	2	10	5	Muy fácil
42	1	10	5	Muy fácil
43	3	10	5	Muy difícil
44	2	12	7	Regular
45	1	10	3	Muy fácil
46	3	10	7	Muy difícil
47	2	10	5	Fácil
48	1	15	5	Muy fácil
49	1	10	7	Muy fácil
50	3	20	10	Difícil
51	3	19	8	Regular
52	2	15	6	Regular
53	2	12	6	Fácil
54	2	17	7	Muy fácil
55	3	27	17	Difícil
56	2	20	10	Regular
57	2	20	10	Muy fácil
58	3	36	10	Regular
59	3	20	0	Regular
60	1	10	5	Muy fácil
61	1	13	4	Muy fácil

## ANEXOS

62	1	15	5	Muy fácil
63	3	25	17	Regular
64	1	12	4	Fácil
65	2	14	9	Fácil
66	1	18	3	Muy fácil
67	0			Muy fácil
68	0			Regular
69	3	16	2	Regular
70	0			Fácil
71	1	12	3	Muy fácil
72	3	20	9	Regular
73	2	12	5	Difícil
74	3	11	4	Fácil
75	1	5	1	Fácil
76	3	10	2	Regular
77	1	15	9	Muy fácil
78	3	9	4	Difícil
79	3	20	10	Difícil
80	2	12	7	Fácil
81	3	17	7	Muy fácil
82	1	16	6	Muy difícil
83	3	18	5	Difícil
84	0			Muy fácil
85	3	20	9	Regular
86	2	16	5	Difícil
87	1	8	1	Fácil
88	2	6	0	Muy fácil
89	2	14	9	Muy difícil
90	0			Fácil
91	2	7	0	Fácil
92	1	10	2	Muy fácil
93	2	13	1	Fácil
94	0			Muy fácil
95	1	10	2	Muy fácil
96	3	16	7	Muy difícil

## ANEXOS

97	0	0	0	Muy fácil
98	2	26	2	Muy fácil
99	3	14	3	Muy difícil
100	2	17	5	Fácil
101	1	14	6	Muy fácil
102	2	12	3	Muy difícil
103	2	20	2	Fácil
104	2	2	0	Fácil
105	3	10	5	Fácil
106	3	5	0	Regular
107	2	20	5	Regular
108	2	18	2	Regular
109	2	15	2	Regular
110	2	13	9	Fácil
111	2	10	1	Fácil
112	3	18	6	Difícil
113	2	9	2	Fácil
114	1	15	1	Muy fácil
115	4	20	13	Difícil
116	4	20	7	Fácil
117	1	12	2	Muy fácil
118	2	12	1	Regular
119	2	18	0	Fácil
120	1	16	4	Muy fácil
121	1	15	4	Regular
122	2	16	6	Fácil
123	2	15	3	Muy fácil
124	4	25	9	Difícil
125	2	18	3	Fácil
126	2	10	5	Muy fácil
127	4	17	5	Muy difícil
128	1	16	2	Fácil
129	1	10	3	Muy fácil
130	3	12	10	Muy difícil
131	2	20	6	Regular

## ANEXOS

132	1	9	1	Muy fácil
133	3	26	17	Muy difícil
134	1	10	1	Fácil
135	1	17	0	Muy fácil
136	3	17	5	Muy difícil
137	1	10	0	Muy fácil
138	1	13	0	Muy fácil
139	1	17	5	Difícil
140	2	10	3	Fácil
141	1	8	1	Muy fácil
142	5	45	10	Difícil
143	1	13	2	Muy fácil
144	1	16	2	Muy fácil
145	3	22	10	Muy difícil
146	1	17	3	Fácil
147	1	6	0	Fácil
148	3	27	11	Muy difícil
149	1	14	2	Fácil
150	2	15	6	Regular
151	3	19	6	Muy difícil
152	2	14	2	Fácil
153	1	13	2	Muy fácil
154	2	23	8	Muy difícil
155	1	18	7	Regular
156	2	26	13	Muy fácil
157	4	23	13	Muy difícil
158	2	20	5	Regular
159	3	27	16	Difícil
160	3	21	9	Muy difícil
161	1	13	5	Muy difícil
162	1	26	11	Muy fácil
163	3	15	15	Muy difícil
164	2	17	9	Regular
165	1	21	11	Muy fácil
166	1	11	1	Difícil

## ANEXOS

167	1	9	19	Regular
168	2	20	15	Fácil
169	4	30	15	Muy difícil
170	1	10	1	Fácil
171	2	20	3	Muy fácil
172	1	15	0	Difícil
173	2	17	5	Fácil
174	1	17	7	Muy fácil
175	3	11	13	Difícil
176	2	16	4	Fácil
177	1	17	1	Muy fácil
178	3	21	11	Difícil
179	2	13	6	Fácil
180	1	13	4	Muy fácil
181	3	24	13	Muy difícil
182	2	17	7	Regular
183	1	22	6	Muy fácil
184	3	26	14	Muy difícil
185	2	15	7	Regular
186	0			Regular
187	3	22	3	Muy difícil
188	1	12	3	Regular
189	1	12	7	Muy fácil
190	3	13	15	Muy difícil
191	2	17	4	Fácil
192	1	12	2	Muy fácil
193	3	17	10	Muy difícil
194	2	23		Regular
195	3	20	7	Fácil
196	3	20	4	Difícil
197	2	19	8	Fácil
198	2	12	3	Muy fácil
199	2	24	16	Difícil
200	3	15	7	Regular
201	1	17	6	Difícil

## ANEXOS

202	3	20	10	Difícil
203	1	12	3	Regular
204	1	15	6	Muy fácil
205	3	20	15	Difícil
206	1	11	3	Fácil
207	1	8	2	Muy fácil
208	2	17	5	Regular
209	1	13	3	Fácil
210	1	17	0	Muy fácil
211	3	27	11	Muy difícil
212	3	17	3	Difícil
213	2	9	1	Muy fácil
214	4	17	8	Muy difícil
215	1	20	7	Fácil
216	1	16	1	Muy fácil
217	2	18	7	Muy difícil
218	1	16	3	Regular
219	2	18	7	Regular
220	1	11	0	Muy fácil
221	2	10	6	Muy fácil
222	3	18	4	Fácil
223	2	21	16	Fácil
224	1	10	0	Muy fácil
225	2	26	11	Fácil
226	3	17	2	Muy fácil
227	2	21	10	Fácil
228	1	15	6	Fácil
229	2	20	14	Fácil
230	3	15	9	Muy fácil
231	2	17	10	Muy fácil
232	2	16	9	Muy fácil
233	3	19	8	Muy fácil
234	1	21	16	Muy fácil
235	4	27	14	Fácil
236	1	16	0	Muy fácil

## ANEXOS

237	3	18	6	Muy fácil
238	2	12	3	Muy fácil
239	1	19	5	Muy fácil
240	2	14	9	Muy fácil
241	2	15	6	Muy fácil
242	2	18	7	Muy fácil
243	1	16	4	Muy fácil
244	1	21	10	Muy fácil
245	1	12	5	Muy fácil
246	1	19	11	Muy fácil
247	1	26	11	Fácil
248	2	18	17	Fácil
249	2	26	19	Fácil
250	2	21	17	Muy fácil
251	4	22	5	Muy fácil
252	2	17	6	Muy fácil
253	3	10	3	Muy fácil
254	2	26	15	Muy fácil
255	2	20	17	Muy fácil
256	3	17	7	Muy fácil
257	1	16	2	Muy fácil
258	3	18	3	Fácil
259	2	18	7	Muy fácil
260	1	31	19	Muy fácil
261	2	36	18	Muy fácil
262	1	28	10	Muy fácil
263	4	20	27	Regular
264	3	32	7	Muy fácil
265	1	20	10	Muy fácil
266	1	15	5	Muy fácil
267	1	17	19	Muy fácil
268	1	35	27	Muy fácil
269	2	17	7	Fácil
270	3	30	15	Fácil
271	4	17	6	Regular



## ANEXOS

272	1	24	16	Regular
273	2	26	4	Muy fácil
274	2	15	7	Muy fácil
275	4	31	26	Muy fácil
276	3	26	12	Muy fácil
277	1	17	16	Regular
278	1	17	7	Regular
279	1	11	4	Regular
280	2	17	6	Fácil
281	1	15	7	Fácil
282	2	20	16	Regular
283	1	20	3	Fácil
284	1	12	2	Muy fácil
285	3	17	7	Regular
286	1	28	16	Regular
287	2	20	21	Regular
288	3	18	7	Fácil
289	1	15	4	Fácil
290	3	26	15	Regular
291	2	26	15	Regular
292	2	26	11	Difícil
293	1	17	6	Fácil
294	2	23	10	Fácil
295	2	20	7	Regular
296	3	32	16	Regular
297	2	21	16	Muy fácil
298	2	30	10	Difícil
299	4	20	21	Difícil
300	1	16	7	Regular
301	2	19	11	Fácil
302	2	16	11	Fácil
303	3	24	15	Regular
304	2	16	10	Regular
305	8	15	5	Muy fácil
306	3	18	9	Fácil

## ANEXOS

307	5	10	3	Fácil
308	2	17	20	Regular
309	1	9	6	Fácil
310	3	27	10	Regular
311	2	21	13	Regular
312	1	20	7	Fácil
313	4	35	19	Difícil
314	1	17	4	Fácil
315	2	24	13	Regular
316	3	36	18	Fácil
317	2	28	15	Fácil
318	3	24	11	Fácil
319	2	35	9	Fácil
320	2	22	14	Regular
321	2	15	4	Fácil
322	2	23	6	Fácil
323	4	32	25	Regular
324	2	20	10	Fácil
325	1	22	12	Regular
326	4	25	18	Muy difícil
327	1	29	24	Muy difícil
328	4	37	21	Muy difícil
329	2	30	16	Difícil
330	2	17	17	Muy difícil
331	3	26	13	Difícil
332	2	30	20	Muy difícil
333	2	26	6	Regular
334	2	31	16	Regular
335	3	37	18	Muy difícil
336	1	30	20	Muy difícil
337	2	15	10	Muy difícil
338	3	25	14	Muy difícil
339	3	30	14	Muy difícil
340	2	26	10	Regular
341	2	20	6	Difícil

## ANEXOS

342	3	28	16	Muy difícil
343	2	12	19	Muy difícil
344	1	20	21	Difícil
345	3	29	25	Muy difícil
346	1	17	19	Muy difícil
347	3	36	16	Muy difícil
348	2	19	2	Difícil
349	2	11	2	Difícil
350	2	21	17	Regular
351	3	25	17	Muy difícil
352	2	17	10	Muy difícil
353	4	40	24	Muy difícil
354	2	20	18	Muy difícil
355	3	27	15	Muy difícil
356	3	37	13	Muy difícil
357	1	20	18	Regular
358	2	20	10	Muy difícil
359	2	37	13	Muy difícil
360	3	36	21	Muy difícil
361	2	26	14	Muy difícil
362	1	17	12	Muy difícil
363	3	26	3	Muy difícil
364	2	20	10	Muy difícil
365	3	35	18	Muy difícil
366	1	6	3	Regular
367	2	21	16	Muy difícil
368	4	32	19	Muy difícil
369	1	24	15	Muy difícil
370	3	26	7	Muy difícil
371	2	18	6	Difícil
372	2	20	10	Regular
373	1	18	9	Difícil
374	2	21	13	Regular
375	3	25	19	Difícil

## ANEXOS

Alumno	<b>18. Se sintió cómodo usando esta herramienta en la creación de sus requisitos</b>	<b>19. Cree usted que la herramienta que usó es la adecuada para el proceso de creación de requisitos</b>
1	Sí	Sí
2	Regular	No
3	Sí	No
4	Sí	Sí
5	Regular	Sí
6	Sí	Sí
7	Sí	Sí
8	Regular	No
9	Sí	Sí
10	No	No
11	Sí	Sí
12	Regular	Sí
13	Regular	No
14	Sí	Sí
15	Regular	No
16	Sí	Sí
17	No	No
18	Sí	Sí
19	Sí	Sí
20	No	No
21	Sí	Sí
22	Sí	Sí
23	No	No
24	Sí	Sí
25	Sí	No
26	No	No
27	Sí	Sí
28	Sí	Sí
29	Regular	No
30	Sí	Sí
31	Sí	Sí

## ANEXOS

32	Sí	Sí
33	Sí	Sí
34	No	No
35	Regular	Sí
36	Sí	Sí
37	No	No
38	No	No
39	Sí	Sí
40	No	Sí
41	Sí	Sí
42	Sí	Sí
43	No	No
44	Sí	Sí
45	Sí	Sí
46	No	No
47	Regular	Sí
48	Sí	Sí
49	Sí	Sí
50	Regular	No
51	Regular	Sí
52	Regular	No
53	Sí	Sí
54	Sí	Sí
55	No	No
56	Regular	No
57	Sí	Sí
58	No	No
59	Sí	Sí
60	Sí	Sí
61	Sí	Sí
62	Sí	Sí
63	Regular	No
64	Sí	Sí
65	Sí	Sí
66	Sí	Sí

## ANEXOS

67	Sí	Sí
68	Sí	Sí
69	Regular	No
70	Regular	Sí
71	Sí	Sí
72	Regular	No
73	No	No
74	Sí	Sí
75	Sí	Sí
76	Regular	No
77	Sí	Sí
78	No	Sí
79	Regular	Sí
80	Sí	Sí
81	Sí	Sí
82	No	No
83	No	No
84	Sí	Sí
85	Regular	Sí
86	No	No
87	Sí	Sí
88	Sí	Sí
89	No	No
90	Sí	Sí
91	Sí	Sí
92	Sí	Sí
93	Sí	Sí
94	Sí	Sí
95	Sí	Sí
96	No	No
97	Sí	Sí
98	Sí	Sí
99	No	No
100	Sí	Sí
101	Sí	Sí

## ANEXOS

102	Regular	No
103	Sí	Sí
104	Sí	Sí
105	Sí	Sí
106	Regular	No
107	Regular	No
108	Regular	Sí
109	Regular	No
110	Sí	Sí
111	Sí	Sí
112	No	No
113	Sí	Sí
114	Sí	Sí
115	No	No
116	Sí	Sí
117	Sí	Sí
118	Regular	No
119	Sí	Sí
120	Sí	Sí
121	Regular	No
122	Sí	Sí
123	Sí	Sí
124	No	No
125	Sí	Sí
126	Sí	Sí
127	No	No
128	Sí	Sí
129	Sí	Sí
130	No	No
131	Regular	No
132	Sí	Sí
133	No	No
134	Sí	Sí
135	Sí	Sí
136	No	No

## ANEXOS

137	Sí	Sí
138	Sí	Sí
139	No	No
140	Sí	Sí
141	Sí	Sí
142	No	No
143	Sí	Sí
144	Sí	Sí
145	No	No
146	Sí	Sí
147	Sí	Sí
148	No	No
149	Sí	Sí
150	Regular	Sí
151	No	No
152	Sí	Sí
153	Sí	Sí
154	No	No
155	Regular	No
156	Sí	Sí
157	No	No
158	Regular	Sí
159	Regular	No
160	Regular	No
161	Sí	Sí
162	Sí	Sí
163	No	No
164	Regular	No
165	Sí	Sí
166	No	No
167	Regular	Sí
168	Sí	Sí
169	Regular	No
170	Sí	Sí
171	Sí	Sí



## ANEXOS

172	No	No
173	Sí	Sí
174	Sí	Sí
175	No	No
176	Sí	Sí
177	Sí	Sí
178	No	No
179	Sí	Sí
180	Sí	Sí
181	No	No
182	Sí	Sí
183	Sí	Sí
184	No	No
185	Regular	No
186	Sí	Sí
187	No	No
188	Regular	Sí
189	Sí	Sí
190	No	No
191	Sí	Sí
192	Sí	Sí
193	No	No
194	Regular	Sí
195	Sí	Sí
196	Regular	No
197	Sí	Sí
198	Sí	Sí
199	No	No
200	Sí	Sí
201	Sí	Sí
202	No	No
203	Regular	Sí
204	Sí	Sí
205	No	No
206	Sí	Sí

## ANEXOS

207	Sí	Sí
208	Regular	No
209	Sí	Sí
210	Sí	Sí
211	No	No
212	No	No
213	Sí	Sí
214	No	No
215	Sí	Sí
216	Sí	Sí
217	No	No
218	Regular	No
219	Regular	No
220	Sí	Sí
221	Sí	Sí
222	Sí	Sí
223	Sí	Sí
224	Sí	Sí
225	Sí	Sí
226	Sí	Sí
227	Sí	Sí
228	Sí	Sí
229	Sí	Sí
230	Sí	Sí
231	Sí	Sí
232	Sí	Sí
233	Sí	Sí
234	Sí	Sí
235	Sí	Sí
236	Sí	Sí
237	Sí	Sí
238	Sí	Sí
239	Sí	Sí
240	Sí	Sí
241	Sí	Sí

## ANEXOS

242	Sí	Sí
243	Sí	Sí
244	Sí	Sí
245	Sí	Sí
246	Sí	Sí
247	Sí	Sí
248	Sí	Sí
249	Sí	Sí
250	Sí	Sí
251	Sí	Sí
252	Sí	Sí
253	Sí	Sí
254	Sí	Sí
255	Sí	Sí
256	Sí	Sí
257	Sí	Sí
258	Regular	Sí
259	Sí	Sí
260	Sí	Sí
261	Sí	Sí
262	Sí	Sí
263	Regular	Sí
264	Sí	Sí
265	Sí	Sí
266	Sí	Sí
267	Sí	Sí
268	Sí	Sí
269	Sí	Sí
270	Sí	Sí
271	Regular	Sí
272	Sí	Sí
273	Sí	Sí
274	Sí	Sí
275	Regular	Sí
276	Sí	Sí

## ANEXOS

277	Regular	Sí
278	Regular	No
279	Sí	Sí
280	Regular	Sí
281	Sí	Sí
282	Regular	Sí
283	Sí	Sí
284	Sí	Sí
285	Sí	Sí
286	Regular	Sí
287	Regular	Sí
288	Sí	Sí
289	Sí	Sí
290	Regular	Sí
291	Regular	No
292	No	No
293	Sí	Sí
294	Sí	Sí
295	Sí	Sí
296	Regular	Sí
297	Sí	Sí
298	No	Sí
299	No	No
300	Regular	Sí
301	Sí	Sí
302	Sí	Sí
303	Regular	Sí
304	Regular	No
305	Sí	No
306	Sí	Sí
307	Sí	Sí
308	Regular	No
309	Sí	Sí
310	Regular	Sí
311	Regular	Sí

## ANEXOS

312	Sí	Sí
313	No	No
314	Sí	Sí
315	Regular	No
316	Sí	Sí
317	Sí	Sí
318	Sí	Sí
319	Sí	Sí
320	Sí	Sí
321	Sí	Sí
322	Sí	Sí
323	Sí	Sí
324	Sí	Sí
325	No	No
326	No	No
327	No	No
328	No	No
329	No	No
330	No	No
331	No	No
332	No	No
333	Sí	Sí
334	Sí	Sí
335	No	No
336	No	No
337	No	No
338	No	No
339	No	No
340	Sí	No
341	Sí	No
342	No	No
343	No	No
344	No	No
345	No	No
346	No	No

## ANEXOS

347	No	No
348	No	No
349	No	No
350	Regular	Sí
351	No	No
352	No	No
353	No	No
354	No	No
355	No	No
356	No	No
357	Regular	No
358	No	No
359	No	No
360	No	No
361	Regular	No
362	No	No
363	No	No
364	No	No
365	No	No
366	Regular	No
367	No	No
368	No	No
369	No	No
370	No	No
371	Regular	No
372	Regular	No
373	No	No
374	Regular	No
375	Sí	Sí

## **7.4 Anexo 4: Contenido del trabajo**

# **ANÁLISIS Y DISEÑO DE SISTEMAS**

## **Contenido del documento**

### **Introducción**

- Descripción de la empresa
- Definición del problema
- Alcance del sistema informático
- Diccionario de datos

### **Características generales**

- El modelo de procesos de información del área de trabajo

### **Modelo del sistema**

- **Modelo de casos de uso**
- **Modelo conceptual**

### **Requisitos**

- **Requisitos funcionales**
- **Requisitos no funcionales**

### **Diseño de la arquitectura de software**

- **Arquitectura lógica**
- **Arquitectura física**

### **Especificación del diseño detallado**

- **Diseño de la interfaz y reportes**
- **Diseño de la base de datos**

## 7.5 Anexo 5: Análisis Estadístico

### 7.5.1 Hipótesis de calidad

#### 7.5.1.1 Análisis de varianza de las medias de notas

Sea  $x_{ij}$  la NotaRequisito que obtiene el alumno  $j$ -ésimo que ha empleado la herramienta  $i=1,2,3$ . Sea  $\mu_1$  la nota media poblacional en NotaRequisito por usarla herramienta 1,  $\mu_2$  la nota media por usar la herramienta 2, y  $\mu_3$  la nota media por usar la herramienta 3. Sea  $\bar{x}_i$  la nota media muestral obtenida a partir de los datos de los 375 alumnos. Con la muestra se han obtenido las medias muestrales, tamaños y desviaciones estándar, presentadas en la tabla siguiente:

Herramienta	Cantidad Alumnos	Media	Desviación Estándar
1	125	10.2960	2.67914
2	124	12.5806	2.65678
3	126	13.9683	2.61667
Total	375	12.2853	3.04883

Se desea analizar si las diferencias entre las medias muestrales es significativa; es decir, si se observan porque las medias poblacionales de cada herramienta  $\mu_i$  son diferentes. Con este fin se realiza un contraste de medias de los tres grupos mediante la técnica ANOVA (Análisis de la varianza). El contraste de hipótesis que se va a realizar es:

$$H_0: \mu_1 = \mu_2 = \mu_3$$

$$H_1: \text{alguna media es diferente}$$

La tabla ANOVA para NotaRequisito por cada herramienta la vemos en la tabla siguiente:

Fuente	Suma de cuadrados	Grados de libertad	Cuadrado medio	F	P-Valor
Intergrupo	862.36	2	431.177	61.36	0.00000
Intragrupo	2614.11	372	7.02719		
Total	3476.47	374			

Usando un nivel de significación habitual de  $\alpha=0.05$ , vemos que el P-Valor (probabilidad de haber obtenido el resultado encontrado asumiendo que la hipótesis es nula es cierta) del contraste es mucho menor, por lo que rechazamos  $H_0$ . Los datos muestran mucha evidencia en contra de  $H_0$ . Las diferencias encontradas son altamente significativas. La herramienta explica el 24.8% ( $R^2 = 0.248$ ) de la variabilidad de la nota de requisito.

#### 7.5.1.2 Influencia de otras variables

Para ver la influencia del resto de variables (sede, tamaño del grupo, índice académico) se ha procedido a realizar una regresión múltiple. En ella, las variables



## ANEXOS

cualitativas (sede, herramienta) se han reemplazado por variables binarias. Y se han creado tres variables binarias para la herramienta, y otras tres para sede. Por ejemplo, la variable (Herramienta=1) vale 1 para los alumnos que utilizaron la herramienta 1 y 0 en el resto. Asimismo, en una regresión, no pueden incluirse todas las variables binarias en las que se descompone una variable cualitativa. Debe excluirse una, al resultar redundantes. Por ejemplo, si para un alumno sabemos que (Herramienta=1) vale 0 y (Herramienta=2) vale también 0, por descarte necesariamente se cumplirá que (Herramienta=3) valdrá 1 (tiene que haber utilizado una). Si introducimos las tres se produce un error numérico. Sobre la metodología estadística para analizar modelos de regresión con variables binarias se puede ver, por ejemplo, en Peña (Peña, 2010).

A continuación, se muestra un resumen de los resultados proporcionados por Statgraphics para el análisis de regresión múltiple que explique NotaReq en función de un conjunto de factores:

Dependent variable: NotaReq

Independent variables:

(Sede=2)

(Sede=3)

(Herramienta=1)

(Herramienta=2)

IndAcad

TamGrup

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	10,6212	1,27765	8,3131	0,0000
Sede=2	0,984762	0,330342	2,98103	0,0031
Sede=3	0,564335	0,379276	1,48793	0,1376
Herramienta=1	-3,50194	0,335201	-10,4473	0,0000
Herramienta=2	-1,38835	0,346581	-4,00584	0,0001
IndAcad	0,294407	0,0796466	3,69642	0,0003
TamGrup	-0,136549	0,20322	-0,671929	0,5021

R-squared = 29,2369 percent

R-squared (adjusted for d.f.) = 28,0831 percent

Standard Error of Est. = 2,58553

La ecuación ajustada es:

$$\begin{aligned} \text{NotaReq} = & 10,6212 + 0,984762 * \text{Sede}=2 + 0,564335 * \text{Sede}=3 - \\ & 3,50194 * \text{Herramienta}=1 - 1,38835 * \text{Herramienta}=2 + 0,294407 * \text{IndAcad} - \\ & 0,136549 * \text{TamGrup} \end{aligned}$$

Aunque esta ecuación ajustada no es la definitiva (posteriormente se descartarán las variables no significativas), en este momento vamos a asumirla para exponer un

## ANEXOS

ejemplo que facilite el entendimiento del uso de las variables binarias en las variables cualitativas.

Como mencionamos las variables cualitativas (sede, herramienta) se han reemplazado por variables binarias (Sede1, Sede2, Sede3, Herramienta1, Herramienta2 y Herramienta3). Índice Académico (IndAcad) y Tamaño de Grupo (TamGrup) son cuantitativas. Por ejemplo:

Ejemplo: para un colectivo de alumnos que cumplan Sede = 2 y Herramienta = 1 tendríamos que las variables binarias utilizadas tomarían los siguientes valores:

Sede1 = 0	Herramienta1 = 1
Sede2 = 1	Herramienta2 = 0
Sede3 = 0	Herramienta3 = 0

Y si, por ejemplo, también asumimos que este colectivo de alumnos tiene IndAcad = 12 y TamGrup = 4, entonces, reemplazando los valores de cada variable en la ecuación de regresión mostrada anteriormente, resulta:

$$\text{NotaReq} = 10,6212 + 0,984762*(1) + 0,564335*(0) - 3,50194*(1) - 1,38835*(0) + 0,294407*(12) - 0,136549*(4)$$

Operando queda:

$$\begin{aligned}\text{NotaReq} &= 10,6212 + 0,984762*(1) - 3,50194*(1) + 0,294407*(12) - 0,136549*(4) \\ &= 11.09071\end{aligned}$$

Este resultado nos indica que el colectivo de alumnos que utilizaron la herramienta 1, que pertenecían a la Sede 2, de índice académico 12 y que pertenecían a grupos de 4 personas, tiene una nota media de 11.09071.

Para analizar la significancia de cada una de las variables del modelo utilizaremos el p-valor. Los p-valores nos indican si cada variable tiene un aporte significativo para explicar el valor medio de NotaReq, si se mantiene el resto de variables en la ecuación. Una variable significativa significa que debe mantenerse en el modelo, mientras que si no es significativa significa que puede eliminarse. Estadísticamente, una variable es significativa cuando sus p-valor es menor al nivel de significación ( $\alpha$ ) que se utiliza, que habitualmente es  $\alpha=0.05$ . De esta forma, los p-valores son el resultado de contrastes de significatividad individual de cada variable

Vemos que hay variables con p-valores  $>0.05$ , y que por tanto no resultan significativas. Esas variables es aconsejable eliminarlas. Como los p-valores sólo nos dicen si las variables son significativas si el resto del modelo se mantiene, hemos de eliminarlas una a una. En cada iteración de este proceso se vuelven a valorar los p-valores, y se decide sobre la siguiente variable a eliminar (en la literatura estadística existen

## ANEXOS

diferentes estrategias para seleccionar las variables de un modelo de regresión, ver (Peña, 2010)). Con esta idea, eliminamos TamGrup (el tamaño de grupo es irrelevante para explicar esta nota TamGrup=0,5021 >  $\alpha=0.05$ ).

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	10,0916	1,00482	10,0432	0,0000
Sede=2	1,01366	0,327288	3,09713	0,0021
Sede=3	0,61217	0,372258	1,64448	0,1009
Herramienta=1	-3,49428	0,334758	-10,4382	0,0000
Herramienta=2	-1,36449	0,344502	-3,96077	0,0001
IndAcad	0,287544	0,0789302	3,64301	0,0003

En el nuevo modelo resultante vemos que la variable (Sede=3) es irrelevante al no ser significativa. Esto quiere decir que la nota media de los alumnos de Sede=3 no difieren de la de la Sede=1, que es la que no aparece en el modelo. Tras eliminarla, se obtiene:

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	10,8495	0,894947	12,1231	0,0000
Sede=2	0,827751	0,307851	2,6888	0,0075
Herramienta=1	-3,61639	0,32717	-11,0535	0,0000
Herramienta=2	-1,48038	0,337993	-4,3799	0,0000
IndAcad	0,24582	0,0749132	3,28141	0,0011

R-squared = 28,6308 percent

R-squared (adjusted for d.f.) = 27,8592 percent

La ecuación ajustada es:

$$\text{NotaReq} = 10,8495 + 0,827751 \cdot \text{Sede=2} - 3,61639 \cdot \text{Herramienta=1} - 1,48038 \cdot \text{Herramienta=2} + 0,24582 \cdot \text{IndAcad}$$

Ahora todas las variables son significativas. El modelo explica el 28.6% de la variabilidad de la nota de requisito. Sólo la herramienta explicaba el 24.8%, por tanto, la sede y el índice académico sólo explican el 3.8%. La principal variable explicativa es la herramienta.

Si incluimos la variable salón, utilizando binarias por cada salón, sólo resulta ser significativa el salón nº 9. El resto no lo son. El modelo final sería:

Dependent variable: NotaReq

Independent variables:

(Herramienta=1)

(Herramienta=2)

(Salón=9)

(Sede=2)

IndAcad

## ANEXOS

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	11,0712	0,889205	12,4506	0,0000
Herramienta=1	-3,83953	0,332764	-11,5383	0,0000
Herramienta=2	-1,72079	0,344564	-4,99409	0,0000
Salón=9	-2,29668	0,785868	-2,92248	0,0037
Sede=2	0,700848	0,307839	2,27667	0,0234
IndAcad	0,249129	0,0741699	3,3589	0,0009

R-squared = 30,2454 percent

R-squared (adjusted for d.f.) = 29,3002 percent

Standard Error of Est. = 2,56355

La ecuación ajustada es:

$$\text{NotaReq} = 11,0712 - 3,83953 * \text{Herramienta}=1 - 1,72079 * \text{Herramienta}=2 - 2,29668 * \text{Salón}=9 + 0,700848 * \text{Sede}=2 + 0,249129 * \text{IndAcad}$$

Ahora todas las variables son significativas. El modelo explica el 30,2% de la variabilidad de la nota de requisito. Sólo la herramienta explica el 24.8%, por tanto, la sede y el índice académico sólo explican el 3.8%. El 1.4% restante viene explicado por el salón 9, que tiene una nota media significativamente menor en 2.3 puntos. La principal variable explicativa es la herramienta.

Con la ecuación definitiva, mostramos otros ejemplos para un mayor entendimiento.

Ejemplo: para un colectivo de alumnos que usó la herramienta 3, del Salón 9, de la Sede 2 y de Índice Académico 12. Reemplazando los valores de cada variable en la ecuación de regresión mostrada anteriormente, resulta:

$$\text{NotaReq} = 11,0712 - 3,83953 * (0) - 1,72079 * (0) - 2,29668 * (1) + 0,700848 * (1) + 0,249129 * (12)$$

Operando queda:

$$\begin{aligned} \text{NotaReq} &= 11,0712 - 2,29668 * (1) + 0,700848 * (1) + 0,249129 * (12) \\ &= 12.464916 \end{aligned}$$

Este resultado nos indica que el colectivo de alumnos que utilizaron la herramienta 3, del Salón 9, que pertenecían a la Sede 2 y de índice académico 12, tiene una nota media de 12.464916.

Ejemplo: ahora, para un colectivo de alumnos que usó la herramienta 3, que no son del Salón 9, no son de la Sede 2 y de Índice Académico 12. Reemplazando los valores de cada variable en la ecuación de regresión mostrada anteriormente, resulta:

$$\text{NotaReq} = 11,0712 - 3,83953 * (0) - 1,72079 * (0) - 2,29668 * (0) + 0,700848 * (0) + 0,249129 * (12)$$

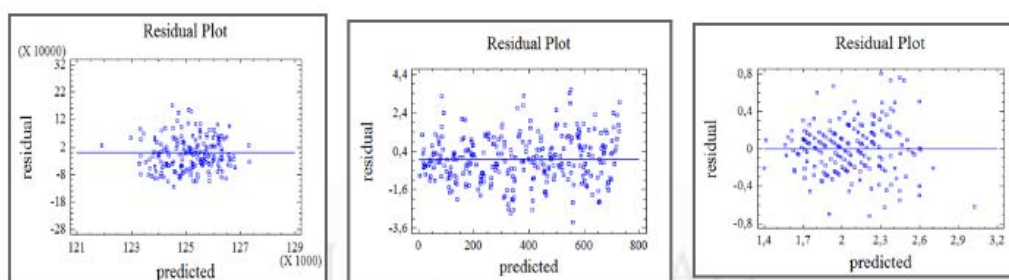
Operando queda:

$$\begin{aligned} \text{NotaReq} &= 11,0712 + 0,249129*(12) \\ &= 14.060748 \end{aligned}$$

Este resultado nos indica que el colectivo de alumnos que utilizaron la herramienta 3, que no son del Salón 9, que no son de la Sede 2 y de índice académico 12, tiene una nota media de 14.060748.

La diagnosis del modelo se puede realizar a través del gráfico de residuos frente a valores previstos (ver (Peña, 2010)). En esta diagnosis se analiza si la relación entre las variables explicativas y la variable respuesta es lineal. Esta linealidad se va a reflejar en la forma que adquiera la nube de puntos de esta figura. Si el modelo lineal utilizado es correcto, la nube de puntos tendrá una distribución uniforme respecto a la línea horizontal que pasa por su centro (en caso de variables discretas los puntos pueden formar líneas que cortan transversalmente a la línea horizontal, lo que no invalida el modelo). Y si el modelo lineal no es correcto, la nube de puntos forma una estructura no lineal, por ejemplo, forma una curva. En la Figura 29 mostramos ejemplos de gráficos de residuos frente a valores previstos correctos e incorrectos (Sánchez, 2017).

## Gráficos de residuos que muestran linealidad (aceptables)



## Gráficos de residuos que no muestran linealidad (NO aceptables)

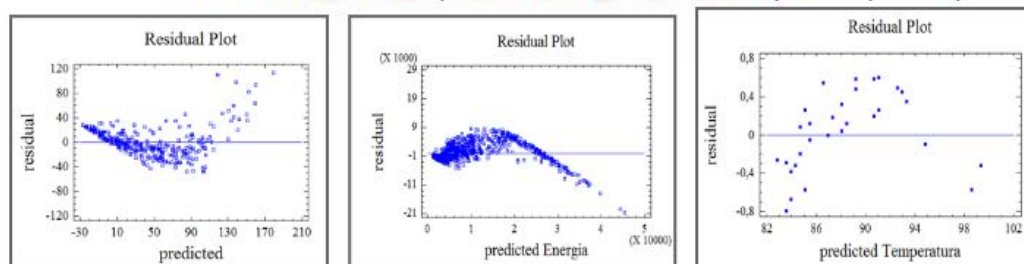
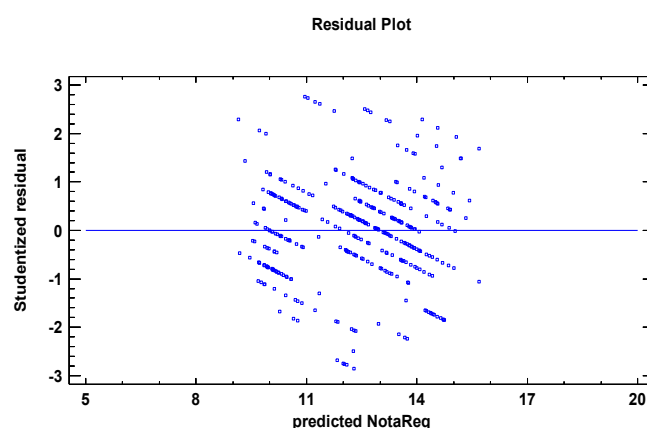


Figura 29 Gráficos de residuos frente a valores previstos correctos e incorrectos (Sánchez, 2017).

Volviendo al análisis estadístico de la tesis, para la diagnosis del modelo se obtiene el gráfico de residuos frente a valores previstos siguiente:



La nube de puntos no muestra ningún patrón anómalo. Por tanto, el modelo lineal utilizado es correcto.

## 7.5.2 Hipótesis de tiempo

Se realizó un análisis de regresión con variables binarias por herramienta, respecto a cada pregunta relevante de la encuesta respecto al tiempo en relación a la herramienta usada.

### 7.5.2.1 Análisis respecto al tiempo promedio para la creación de un requisito

Para realizar la regresión, se han eliminado los datos en los que el tiempo medio para crear un requisito supera los 120 minutos. Son sólo 4 datos que al ser muy grandes pueden tener mucha influencia en los cálculos (sin llegar a cambiar las conclusiones). Estos 4 datos atípicos es muy probable que se deban a error de escritura al especificar el tiempo.

El modelo se ha realizado tomando logaritmos del tiempo medio, pues de esta forma se obtiene un gráfico de residuos más homogéneo. Al tomar logaritmos, la varianza del tiempo medio con cada herramienta es similar, lo que es necesario para que el modelo de regresión sea correcto. La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable:  $\log(T_{\text{medioUnReq}})$

Independent variables:

(Herramienta=1)

(Herramienta=2)

Selection variable:  $(T_{\text{medioUnReq}} < 120)$

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	2,28458	0,0475938	48,0016	0,0000
Herramienta=1	1,52314	0,0678573	22,4462	0,0000
Herramienta=2	0,627256	0,0678573	9,24374	0,0000

## ANEXOS

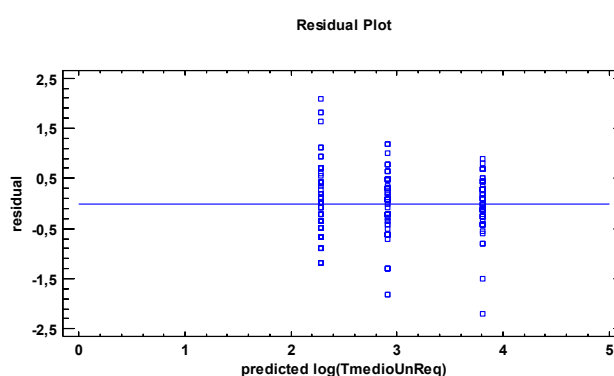
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	145,018	2	72,509	254,05	0,0000
Residual	104,746	367	0,285412		
Total (Corr.)	249,764	369			

R-squared = 58,062 percent

La ecuación ajustada es:

$$\log(\text{TmedioUnReq}) = 2,28458 + 1,52314 \cdot \text{Herramienta} + 0,627256 \cdot \text{Herramienta}^2$$

El gráfico de residuos frente a valores previstos es:



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, el modelo lineal utilizado es correcto.

Hay diferencias significativas entre la herramienta 3 y las otras dos. Los tiempos medios estimados con este modelo son:

- El tiempo medio de realización de un requisito con la herramienta 3 es  $e^{2.28} = 9.98$  minutos.
- Con la herramienta 1 se demora más tiempo. El tiempo medio es  $e^{2.28} \times e^{1.5} = 43.8$  minutos.
- También con la herramienta 2 se demora más tiempo, aunque no tanto como con la 1. El tiempo medio con la herramienta 2 es  $e^{2.28} \times e^{0.63} = 18.3$  minutos.

### 7.5.2.2 Análisis respecto al tiempo promedio para crear toda la lista de requisitos

En este caso, para realizar la regresión, se han eliminado 2 datos atípicos que superan los 2000 minutos. La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable: TtodosReq

Independent variables:

(Herramienta=1)

(Herramienta=2)

Selection variable: TtodosReq<2000

## ANEXOS

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	399,603	20,5905	19,4072	0,0000
Herramienta=1	248,945	29,2365	8,51487	0,0000
Herramienta=2	168,389	29,2964	5,74777	0,0000

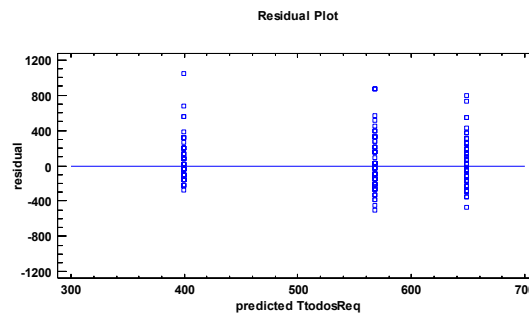
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	4,0394E6	2	2,0197E6	37,81	0,0000
Residual	1,97654E7	370	53420,0		
Total (Corr.)	2,38048E7	372			

R-squared = 16,9688 percent

La ecuación ajustada es:

$$T_{\text{todosReq}} = 399,603 + 248,945 * \text{Herramienta}=1 + 168,389 * \text{Herramienta}=2$$

El gráfico de residuos frente a valores previstos es:



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, el modelo lineal utilizado es correcto.

De nuevo la herramienta 3 es significativamente diferente a las otras dos, con un tiempo medio menor.

La herramienta 3: en media se demora 399.6 minutos. Con la herramienta 2, en media se demora 168 minutos más que con la herramienta 3. Con la herramienta 1, en media se demora 249 minutos más que con la herramienta 3. La herramienta sólo explica el 16% de la demora.

### 7.5.2.3 Análisis respecto a la cantidad de requisitos creados

La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable: NumReq

Independent variables:

(Herramienta=1)

(Herramienta=2)

Selection variable: TtodosReq<2000



## ANEXOS

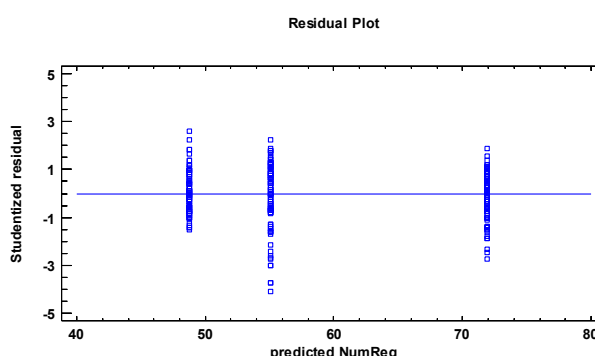
		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	71,9127	1,05161	68,3837	0,0000
Herramienta=1	-23,1869	1,49318	-15,5285	0,0000
Herramienta=2	-16,7078	1,49934	-11,1434	0,0000

R-squared = 41,0627 percent

La ecuación ajustada es:

$$\text{NumReq} = 71,9127 - 23,1869 * \text{Herramienta}=1 - 16,7078 * \text{Herramienta}=2$$

El gráfico de residuos frente a valores previstos es:



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, el modelo lineal utilizado es correcto.

En media, con la herramienta 3 se tienen 71.9 requisitos. Con la herramienta 2, 16 requisitos menos que con la herramienta 3, y con la herramienta 1, 23 requisitos menos que con la herramienta 3. La herramienta explica el 41% de las diferencias en requisitos de cada alumno.

En el resto de relaciones que se analizan en adelante, el  $R^2$  es muy bajo. Quiere decir que se han encontrado diferencias en los valores medios por tipo de herramienta, pero alrededor de esos valores medios hay mucha variabilidad, mucha dispersión.

### 7.5.2.4 Análisis respecto a la modificación de la lista de requisitos

Para realizar el análisis estadístico, a la variable de modificación hay que modelizarla en logaritmos; es decir, como variable dependiente se usa el logaritmo del tiempo (Log(Tmodif)). La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable: log(Tmodif)

Independent variables:

(Herramienta=1)

(Herramienta=2)

## ANEXOS

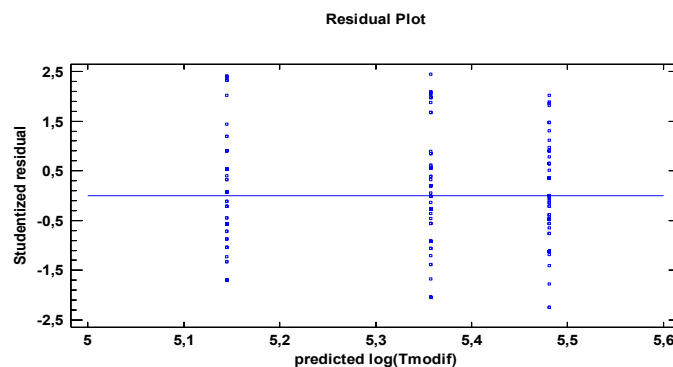
		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	5,14461	0,0555783	92,5652	0,0000
Herramienta=1	0,336613	0,0787566	4,27409	0,0000
Herramienta=2	0,212542	0,0789158	2,69327	0,0074

R-squared = 4,78577 percent

La ecuación ajustada es:

$$\log(T_{\text{modif}}) = 5,14461 + 0,336613 * \text{Herramienta}=1 + 0,212542 * \text{Herramienta}=2$$

El gráfico de residuos frente a valores previstos es:



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, el modelo lineal utilizado es correcto.

El modelo puede escribirse como:

$$T_{\text{modif}} = e^{5.14} \times e^{0.33\text{Herramienta}=1} \times e^{0.33\text{Herramienta}=2}$$

Las diferencias de cada herramienta respecto de la herramienta 3 son significativas. Las estimaciones del tiempo por cada herramienta son las siguientes:

- Por término medio, los usuarios de la herramienta 3 demoran en modificar los requisitos  $e^{5.14} = 170$  minutos
- El tiempo medio que demoran en realizar las modificaciones los usuarios de la herramienta 1 es  $e^{5.14+0.337} = 239$  minutos.
- Para los usuarios de la herramienta 2, el tiempo medio para modificar los requisitos es  $e^{5.14+0.213} = 211$  minutos.

### 7.5.2.5 Análisis respecto a la cantidad de requisitos modificados

La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable: NReqCam

Independent variables:

(Herramienta=1)

(Herramienta=2)

## ANEXOS

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	17,4711	0,659728	26,4822	0,0000
Herramienta=1	3,68093	0,925502	3,97722	0,0001
Herramienta=2	-0,9626	0,938907	-1,02523	0,3059

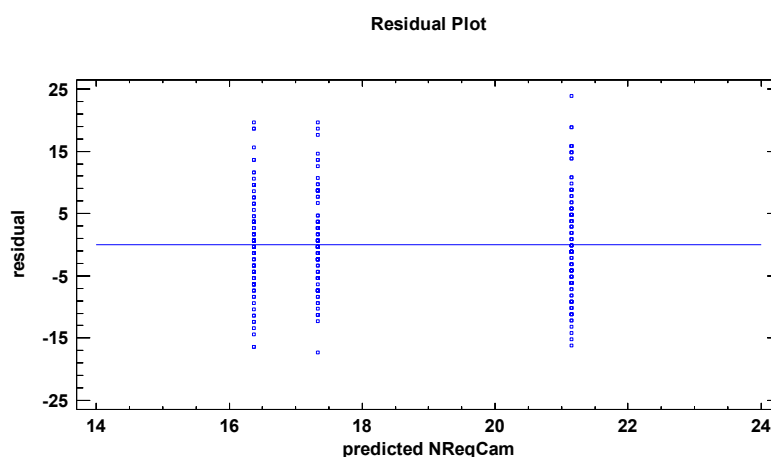
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	1473,09	2	736,547	13,99	0,0000
Residual	19011,8	361	52,6641		
Total (Corr.)	20484,8	363			

R-squared = 7,19114 percent

La ecuación ajustada es:

$$\text{NReqCam} = 17,4711 + 3,68093 * \text{Herramienta}=1 - 0,9626 * \text{Herramienta}=2$$

El gráfico de residuos frente a valores previstos es:



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, nuevamente el modelo lineal utilizado es correcto.

El número de requisitos cambiados también depende de la herramienta. En este caso las herramientas 2 y 3 son similares, no hay diferencias significativas. Con ambas herramientas hacen una media de 17.47 cambios, mientras que con la herramienta 1 se tienen que hacer, por término medio, 3.6 cambios más que con cualquiera de las otras dos herramientas.

Para un mejor análisis se ha construido la variable tiempo por cada cambio=  $T_{\text{modif}}/\text{NReqCam}$ . De esta manera se analiza el tiempo medio que se demora en cambiar un requisito. La salida que proporciona el Statgraphics para este modelo es la siguiente:

## ANEXOS

Dependent variable:  $\log(T_{\text{modif}}/N_{\text{ReqCam}})$

Independent variables:

(Herramienta=1)

(Herramienta=2)

		<i>Standard</i>	<i>T</i>	
<i>Parameter</i>	<i>Estimate</i>	<i>Error</i>	<i>Statistic</i>	<i>P-Value</i>
CONSTANT	2,2998	0,0658897	34,9038	0,0000
Herramienta=1	0,20957	0,0924336	2,26725	0,0240
Herramienta=2	0,293411	0,0939752	3,12222	0,0019

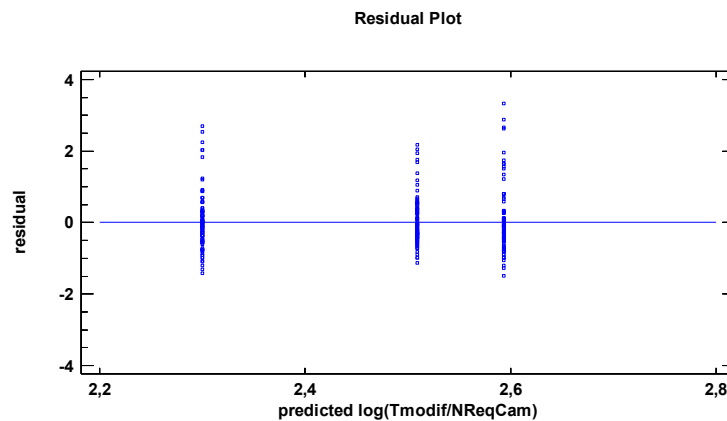
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	5,4707	2	2,73535	5,21	0,0059
Residual	189,114	360	0,525316		
Total (Corr.)	194,584	362			

R-squared = 2,81148 percent

La ecuación ajustada es:

$$\log(T_{\text{modif}}/N_{\text{ReqCam}}) = 2,2998 + 0,20957 \cdot \text{Herramienta}=1 + 0,293411 \cdot \text{Herramienta}=2$$

El gráfico de residuos frente a valores previstos es:



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, nuevamente el modelo lineal utilizado es correcto.

Hay diferencias significativas entre la herramienta 3 y la herramienta 1 o la herramienta 2. El tiempo medio en modificar un requisito es más rápido si se usa la herramienta 3. Las estimaciones de los tiempos medios por cada herramienta son:

- Herramienta 3:  $e^{2.3} = 9.97$  minutos
- Herramienta 1:  $e^{2.3+0.21} = 12.3$  minutos
- Herramienta 2:  $e^{2.3+0.29} = 13.3$  minutos

## ANEXOS

En este caso, la diferencia entre los tiempos de la herramienta 1 y 2 no resultan significativos. Este resultado se obtiene repitiendo la regresión, pero ahora sólo introduciremos variables binarias para las herramientas 2 y 3. Los coeficientes medirán entonces las diferencias de tiempo por requisito entre cada herramienta y la herramienta 1, que es la que no está incluida entre las variables del modelo. Puede verse que la diferencia entre la herramienta 1 y la herramienta 3 sí es significativa (-0.2 con p-valor=0.024), pero entre la 1 y la 2 no lo es (p-valor=0.3691). La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable:  $\log(T_{\text{modif}}/N_{\text{ReqCam}})$

Independent variables:

(Herramienta=3)

(Herramienta=2)

		<i>Standard</i>	<i>T</i>	
<i>Parameter</i>	<i>Estimate</i>	<i>Error</i>	<i>Statistic</i>	<i>P-Value</i>
CONSTANT	2,50937	0,0648269	38,7088	0,0000
Herramienta=3	-0,20957	0,0924336	-2,26725	0,0240
Herramienta=2	0,0838415	0,0932331	0,899268	0,3691

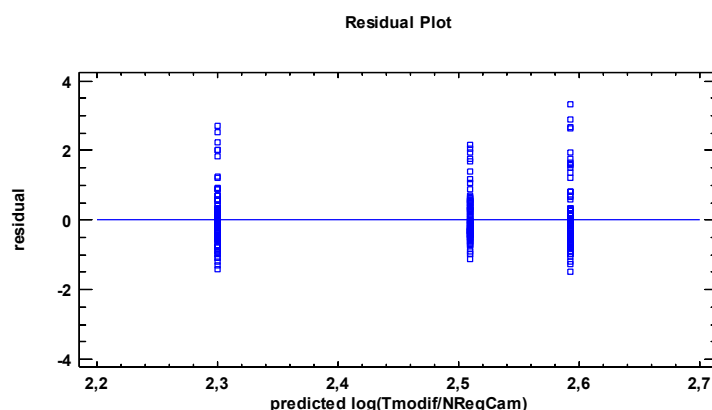
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	5,4707	2	2,73535	5,21	0,0059
Residual	189,114	360	0,525316		
Total (Corr.)	194,584	362			

R-squared = 2,81148 percent

La ecuación ajustada es:

$$\log(T_{\text{modif}}/N_{\text{ReqCam}}) = 2,50937 - 0,20957 * \text{Herramienta}=3 + 0,0838415 * \text{Herramienta}=2$$

**El gráfico de residuos frente a valores previstos es:**



La distribución de los puntos no muestra ningún patrón anómalo. Por tanto, nuevamente el modelo lineal utilizado es correcto.

## 7.5.2.6 Análisis respecto a la cantidad de requisitos eliminados

La salida que proporciona el Statgraphics para este modelo es la siguiente:

Dependent variable: NReqElim

Independent variables:

(Herramienta=1)

(Herramienta=2)

		Standard	T	
Parameter	Estimate	Error	Statistic	P-Value
CONSTANT	7,38017	0,518956	14,2212	0,0000
Herramienta=1	3,24383	0,72802	4,45569	0,0000
Herramienta=2	-0,491276	0,740161	-0,663742	0,5073

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	1009,93	2	504,963	15,50	0,0000
Residual	11731,4	360	32,5872		
Total (Corr.)	12741,3	362			

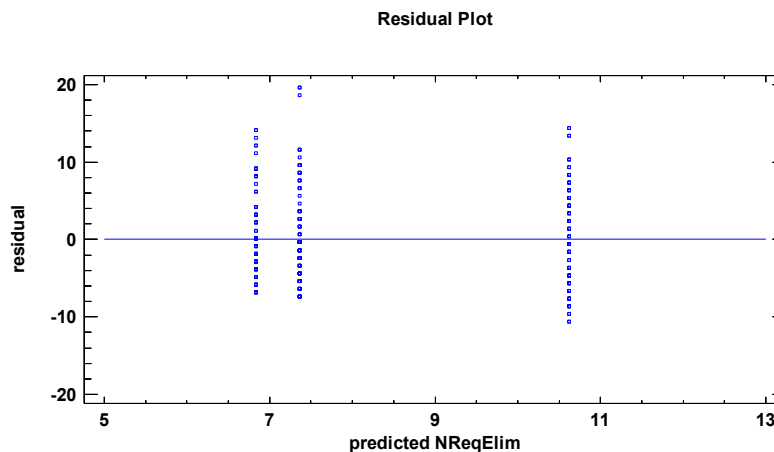
R-squared = 7,92639 percent

La ecuación ajustada es:

$NReqElim = 7,38017 + 3,24383 * Herramienta=1 - 0,491276 * Herramienta=2$

El gráfico de residuos frente a valores previstos es:

La distribución de los puntos no muestra ningún patrón anómalo. Por tanto,



El número de requisitos que tuvieron que ser eliminados tiene también mucha variabilidad de unos alumnos a otros (el  $R^2$  de la regresión es muy bajo), pero hay una diferencia significativa dependiendo de la herramienta. Con la herramienta 1 se tuvieron que eliminar por término medio 3.24 requisitos más que los que usaron la herramienta 2 o la herramienta 3, como se deduce del análisis de regresión. En ella se ve que la herramienta 2 no es significativamente diferente de la 3, que es la que no está incluida (p-valor=0.5073). Función de paso cóncava. La función de paso cóncava es contraria a la convexa, se da siempre y cuando, ante una medida intermedia del indicador, el factor se califica como bajo y para las medidas extremas (pequeñas o grandes) el factor se califica como alto.